NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ECTRACE - THREE DIMENSIONAL ACOUSTIC

RAY TRACE PROGRAM

by

Robert Neal Christianson

Luis Ricardo Erazo

March 1980

Thesis Advisor: A. B. Coppens

80 10 3 082

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A089 937	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ECTRACE - THREE DIMENSIONAL ACOUSTIC RAY TRACE PROGRAM		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; March 1980
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert Neal/Christianson Luis Ricardo/Erazo		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1980
		13. NUMBER OF PAGES 209
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acoustic Computer Model	Synthetic Ocean Bottom Model
Three-Dimensional Ray Trace	Ocean Bottom Modeling
Acoustic Ray Tracing	Bathymetric Curvature Effect
Bathymetric Interference	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Acoustic ray paths in the ocean are known to exhibit significant horizontal deflections after repeated reflections from the bottom. The effect may be quantitatively and qualitatively observed through a ray trace model which permits a change in direction of the vertical plane of propagation as a function of bottom slope and grazing angle. ECTRACE is a family of computer programs which traces a bundle of rays in three dimensions and utilizes bottom depth values as a portion of its input. Included are

DD FORM 1473 EDITION OF 1 NOV 68 IS OBSOLETE
1 JAN 73
(Page 1)
S/N 0102-014-6601

1

20. (Continued)

related programs which develop sea bed models from digitized
bathymetry data or synthetic bathymetry functions.

ECTRACE - Three Dimensional Acoustic Ray Trace
Program

by

Robert Neal Christianson
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1970

Luis Ricardo Erazo
Lieutenant, United States Navy
B.S., Purdue University, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY

from the

NAVAL POSTGRADUATE SCHOOL
March 1980

Authors

_____

_____

Approved by:

_____
                                    Thesis Advisor

_____
                                    Co-Advisor

_____
Chairman, Antisubmarine Warfare Academic Group

_____
                                    Academic Dean

ABSTRACT

Acoustic ray paths in the ocean are known to exhibit significant horizontal deflections after repeated reflections from the bottom. The effect may be quantitatively and qualitatively observed through a ray trace model which permits a change in direction of the vertical plane of propagation as a function of bottom slope and grazing angle. ECTRACE is a family of computer programs which traces a bundle of rays in three dimensions and utilizes bottom depth values as a portion of its input. Included are related programs which develop sea bed models from digitized bathymetry data or synthetic bathymetry functions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I.   INTRODUCTION

ECTRACE is a FORTRAN computer program designed for the investigation of bathymetric effects on horizontal acoustic ray tracing. Its supporting computer programs include routines for constructing a grid model of the sea bed from contour data, ray tracing routines, and associated printed and plotted output. The programs were written for IBM 360/370 series systems with a Versatec plotter and related software. The significant feature of the tracing algorithm is its ability to shift the direction of the vertical plane of propagation after a bottom reflection. A sea bed modeled by triangular facets fitted through three points of a grid cell makes this feature possible with a minimum effect on computation time.

Several models for tracing acoustic rays in the ocean are used in the Navy today. These models are based on a simplified environment because of practical limitations on data and computer run time, and they adequately trace a bundle of rays in a single vertical plane. Some enable intensity calculations at a given distance from the source. TRIMAIN [1], for example, is a model which features range as well as depth dependence of sound speed and accounts for some interaction with irregular bathymetry. These models have a large number of applications in deep ocean cases involving ducts, channels, and convergence zones, and bottom reflections where the plane of the ray path does not exhibit horizontal deflection.

Actual cases of long range propagation over irregular bottoms show any single ray path rarely remains within a single vertical plane. Any application of ray theory to propagation involving bottom interaction must account for a horizontal deflection caused by ray heading changes after repeated reflection from a tilted bottom plane. This effect is most pronounced for long range paths over smoothly sloping surfaces, but any valid ray approximation can be seen to undergo significant heading changes after only a few reflections from an irregular or undulating bottom.

The behavior of sound propagation over troughs, ridges, wedges, and seamounts has been studied analytically by Harrison [2] using normal mode theory and ray invariants. Since actual ocean bottom topography over large area defies practical analytical description, except in the stochastic sense, DeWitt [3] developed numerical techniques for three-dimensional ray tracing in which the bottom consists of triangular facets generated numerically from actual or interpolated bathymetry data. This approximation scheme greatly simplified the problem of calculating the three-dimensional ray path parameters after a bottom reflection and provides a means of constructing reasonably accurate ray traces (for a verification, see Appendix F).

The ECTRACE modeling program is an adaptation of the Dewitt technique for use on an IBM 360/370 series computer. The initial programs may be used to generate synthetic bottom topographical grids or create approximations to real ocean bottom grids from a bathymetric data source file. The

topographical features of the generated grid can be depicted

by a contour plot and a perspective surface plot for compari-

son with the original contours. The primary program is the

ray tracing program which accepts the generated bottom grid

and performs stepwise ray position calculations from user-

selected initial heading and elevation angles. The output

is the printed history of each ray, and a two-dimensional plot

of the ray paths projected on a vertical plane intersecting

the bottom profile along the axis of the ray fan. Addition-

ally, a horizontal plane projection is plotted which may be

overlaid on the appropriate contour plot to correlate the

horizontal curvature effect with grid bathymetry. Punched

card output from several ray tracing runs may be combined in

one independent routine for a comprehensive visual inspection

of acoustic shadow zones in the horizontal plane.

It is very important to note that ECTRACE permits refrac-

tion caused by sound velocity gradients in the vertical plane

only and that the sound velocity in the horizontal plane is

assumed constant within a well defined water mass. All hori-

zontal bending effects revealed by ECTRACE are the result of

bottom reflection only.

With the ability to convert actual bathymetry data to a

matrix of depth values, discrete faceted approximations of a

selected oceanic region can be used for generating ray traces

of operational significance. In addition, qualitative studies

of horizontal ray deflections over idealized bottom configura-

tions are enhanced by the computer-generated plots of ray paths

over easily modifiable synthetic bathymetry. ECTRACE and the

14

augmenting programs are envisioned to serve as a basis for future development of three-dimensional ray modeling and investigation efforts.

II. DESCRIPTION OF ECTRACE AND PERIPHERAL PROGRAMS

This chapter describes each of the programs, their pri-
mary subroutines, and their interrelationship in producing a
ray trace which may be used to investigate the bathymetric
effects on horizontal ray curvature. They are presented in
their intended order of use. Flow Diagram (Fig. 1) and Table
I are provided to assist in understanding program relationships.

A. GENBOT

This program converts contour data into a matrix of depth
values for storage on a permanent device. It also makes a
two dimensional contour plot of the input data and super-
imposes reference latitude and longitude lines (Fig. 2).
The contour plot can be placed underneath the horizontal ray
trace plot of ECTRACE or ECCOM on a light table to aid in
the visual investigation of the ray curvature (see Figs. 3
and 4, and Figs. 5, 6, and 7).

The bathymetry contour data used in this research exists
as a sequence of data sets representing ten by ten degree
regions of the North Atlantic Ocean and were obtained exclu-
sively from a tape compiled in a joint project of the Naval
Research Laboratory and the Mobil Oil Company. In their
original form the data were sufficient only for producing a
computer plot of region contours, thereby reproducing por-
tions of a chart titled "Bathymetry of the Norwegian-Green-
land and Western Barents Sea" [4] (see Fig. 8). Their use

in GENBOT required us to insert the depth values of the contour lines and transfer the results to a separate tape (described in Appendix A).

The following subroutines are called by GENBOT:

1) CORNER calculates boundary latitudes and longitudes for screening only those data needed for interpolation.

2) GEOPLT superimposes reference latitude and longitude lines on the subregion contours.

3) GEODST calculates the forward or inverse solutions of the geodetic triangle to accomplish the above.

4) RAIN 1, RAIN 2, and RAIN 3 [5] were obtained from the SSP3 program library at NPS and are used to perform interpolations on data points for construction of the depth matrix.

B. SYNGEN

This program produces a synthetic bathymetry grid for storage on a permanent device. It may be used to generate the following types of ideal sea bed configurations: wedge, trough, ridge, conical seamount (Fig. 9), sinusoidal undulation, or parabolic basin (Fig. 10). These grids can be used in the ECTRACE program as a depth matrix in the same manner as the grids of actual bathymetry produced by GENBOT. SYNGEN is discussed in detail in Appendix B.

C. G3DP

Program G3PD produces a perspective of any portion of the generated depth matrix through use of the NPS system subroutine CONTUR [4]. Figures 9, 10, and 11 are examples. G3DP uses

job control commands to link with the GRDSCT subroutine of the ECTRACE load module to extract the desired area (hereafter called the working matrix) to be plotted. Appendix C describes the application of G3DP in detail.

D. GRDCHK

This program is used to check the validity of the depth matrix generated by the programs GENBOT and SYNGEN. It produces a contour plot and a vertical plot of user-selected rows of the generated matrix. By comparing consecutive rows the existence and location of anomalous data points may be determined (Figs. 4 and 12). It contains the previously mentioned subroutine GEODST, CORNER, and a version of GEOPLT (called GEOPLR) to draw reference geographic lines for comparison of the two contour plots (GENBOT vs GRDCHK). Refer to Appendix D for additional discussion of GRDCHK.

E. ECTRACE

The ECTRACE program is a package of subroutines which exist in load module form on a permanent storage device in the NPS computer system user library. Its use requires a user supplied calling program which calls subroutine TRACER and contains the necessary job control language (JCL) to link with the module. These procedures are explained in more detail in Chapter IV of this report.

ECTRACE traces rays in three dimensions. The sound speed field is assumed to vary piecewise-linearly with depth only, yet provisions are made to permit simulation of up the three distinct water masses separated discontinuously by vertical

18

fronts. The bathymetry structure, generated by one of the two grid-making programs, is a three dimensional surface modeled by discrete triangular facets fitted through cells of adjacent depth values. Rays are traced as piecewise arc segments each with a radius of curvature dependent on the vertically-structured sound speed gradient.

When the rays undergo a bottom bounce, specular reflection is assumed. Bottom loss values can be calculated from the subprogram FBLOSS (described later), a standard Navy model such as the FACT bottom-loss routine, or a user supplied routine. The trace history supplied by ECTRACE includes the location and characteristics of each ray reversal (surface reflections, bottom bounces, and refractive turning points). The printed data lists the new ray parameters determined after the reversal had occurred. Chapter V explains the printed output in more detail.

The ECTRACE plot product includes a plane view of the ray paths projected on a vertical plane, a plot of sound speed profiles, and a horizontal (plane) view of the ray paths (Figs. 5 and 13). The horizontal view can be overlaid on the appropriate contour plot generated by GENBOT or GRDCHK, as in Figs. 12 and 14.

The following subroutines are used by ECTRACE:

1) Tracer is the primary tracing subroutine which calls all other subroutines of the ECTRACE program and is capable of processing a bundle of rays separated into ray fans. A ray fan is defined here as a number of

rays (user determined) each with a different initial
heading measured clockwise from grid north, but all
with the same initial elevation angle. A positive
elevation angle describes a ray which is pointed toward
increasing depth.

2) GRDSCT reads into core a fixed-dimension section of a
depth matrix file of arbitrary size. The ray trace is
restricted to the boundaries of the core-loaded working
matrix. A ray which departs the working matrix boundaries
is terminated and program control is passed to the next
ray's trace.

3) IDTSUB identifies the three matrix subscript pairs whose
depth values define a plane triangular bottom facet
underneath the ray segment head.

4) DEEP calculates the depth at any horizontal position by
solving for the z-coordinate on a plane fitted through
the three depth matrix values.

5) CONTAC calculates the three-dimensional coordinates of
the intersection of the ray path and triangular bottom
facet.

6) BOANG calculates the grazing angle, elevation angle, and
new heading of a bottom reflected ray.

7) FBLOSS returns bottom loss values in dB as a function
of grazing angle for a bottom reflected ray. These
values are based on NRL geophysical survey data of the
Greenland-Norwegian Sea and Iceland environment collected
in the mid-1970's and have been widely averaged for

convenience. The user has the option of substituting a more specialized function if bottom loss values are critical.

8) IDPROF determines the water mass, and hence the sound speed profile, affecting the ray with each segment iteration.

9) NUPROF initializes calculations for new water mass parameters once the ray has passed the boundary.

10) CHNLIM determines in advance the refractive sound channel limits for each ray from its invariant. This is used to test a ray at its turn-around point and to reveal its type (surface reflected, refracted-surface reflected, bottom reflected-surface reflected) in the printed trace history.

11) ANGPRT is used to print a summary of selected bottom contact in parameters for each ray fan.

12) The following subroutines each have only one calling statement in TRACER. The subroutines they in turn invoke are dependent upon the computer graphic system installed and would require internal modification if exported to a system without Versatec software. For additional information NPS users should consult Ref. 7.

   a) BGNPLT initializes the plotter and draws all borders, titles, and axes.

   b) BDTPLT draws a vertical profile of the sea bed centered along the mean heading of the ray bundle, as shown in the upper plot of Figure 13.

21

c) RNGPLT traces all ray segment projections on the vertical plot.

d) T2DPLT plots the horizontal track of each ray. Each ray fan uses one of thirteen symbols to indicate the bottom bounce positions of each ray. Figure 5 shows some of the available symbols.

e) SSPPLT draws the sound speed profiles.

f) ENDPLT draws the horizontal plot legend and terminates all plotting.

F. ECCOM

Program ECCOM uses the optionally produced punched card output of one or more ECTRACE jobs to draw a composite horizontal plot of several ray bundles, enabling a comprehensive pictorial study of shadow zones and acoustic convergence in the horizontal plane. The composite horizontal plot is designed to portray the rays which emanate from the same source or from multiple sources along a desired track. The ray numbers assigned and punched for each ray on each ECTRACE run assist the user in singling out rays of interest (or noninterest) before combination. Figures 7 and 18 are examples of this combining technique.

Amplifying remarks on these programs and subroutines can be found in the comments in the computer FORTRAN listings. For additional description of plots and computer printed outputs, refer to the appropriate appendices.

III.   PROGRAM LOGIC AND THEORY

ECTRACE traces the three-dimensional paths of a bundle of
rays, one ray at a time. The bundle is divided into fans of
rays of the same initial elevation angle. A separate ray fan
is traced for each fixed increment of initial elevation angle
between the limits ELST and ELEND (input constants). A posi-
tive elevation angle describes a ray pointed downward. The
initial ray headings in each ran fan extend from HDST to HDEND
(input constants), measured in degrees clockwise from grid
north.

A.   BOTTOM AND SOUND FIELD MODELING

Figure III.1 shows the projection of the bottom surface
onto the horizontal x-y plane.
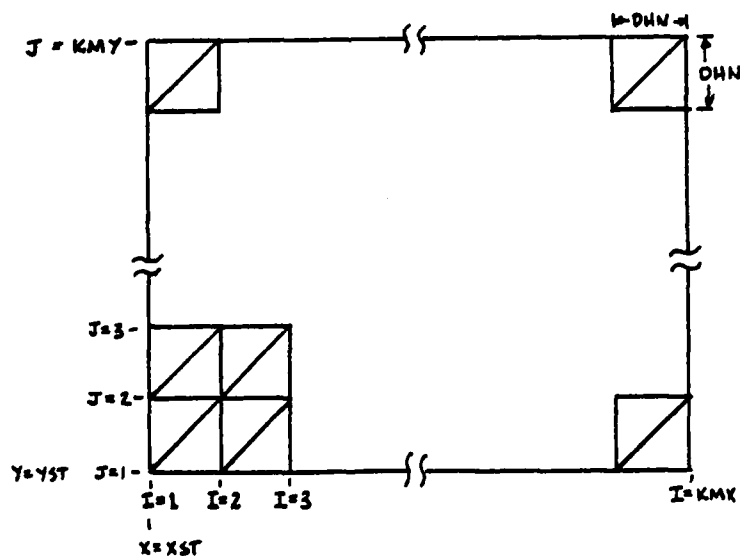
FIGURE III.1   Bottom Surface Projection

The projection of each triangular facet is a right isosceles triangle. The entire surface can be defined by specifying the depth values at the vertices of the projection triangles. ECTRACE stores these depth values (km) in the two-dimensional working array ZB. The parameter DHN (km) represents the spacing between the vertices in both the X (East) and Y (North) directions. Thus, the value of the matrix point ZB(I,J) is the depth at the vertex point with X coordinate (I - 1) * DHN and Y coordinate (J - 1) * DHN, relative to the ZB origin.

The matrix values for ZB are read from a source file stored on a permanent storage device such as a disk pack. A matrix of dimensions 151 by 151 with spacing DHN = km has been used for trial runs.

1. Bottom Contact Point

The subroutine CONTAC locates the coordinates and calculates the depth of the bottom bounce point. The parameters describing locations of the ray segment head and tail are among those passed to the main program. In this subroutine, the ray segment is treated as unrefracted (a straight line), a reasonable approximation for the small distances involved in determining the contact point.

First the horizontal positions of the ray segment head (x,y) called CEE and CNN in CONTAC, and (x_1, y_2) and called CRE and CRN in CONTAC, are checked to see if they both lie in the same projection triangle. If not, an interative routine searches for two points on the segment such that the horizontal coordinates of both points lie inside the same triangular cell and

define the portion of the segment that penetrates the bottom facet. Then the depth of the ray head z, and the depth of the tail z_l are computed.

Figure III.2. illustrates a profile of a cell bottom with a ray intersecting it in the propagation plane. The solution of the triad (x_c, y_c, z_c) identifying the point of bottom contact proceeds as follows:

Using subroutine DEEP (described in Section 3) compute H_1 and H_2, the bottom depth at (x_1, y_1) and (x_2, y_2), respectively. Then compute:

$$\Delta z = z_1 - z_2$$

where (x_c, y_c, z_c) are contact point parameters.

$$z_c = \frac{H_1 z_2 - H_2 z_1}{\Delta z + H_1 - H_2}$$

If $\Delta z \neq 0$, $M = \dfrac{z_c - z_1}{\Delta z}$

If $\Delta z = 0$, $M = \dfrac{z_1 - H_1}{H_1 - H_2}$

Then $x_c = x + M r \sin \phi$ and

$y_c = y + M r \cos \phi$

where ϕ is the ray heading (angle between the y-z plane and the propagation plane).

2. Projection Triangle Vertex Subscripts

The subroutine IDTSUB calculates the I subscripts, IRT, ILT, IVR and the J subscripts JBT, JTP, JVR for the vertices of
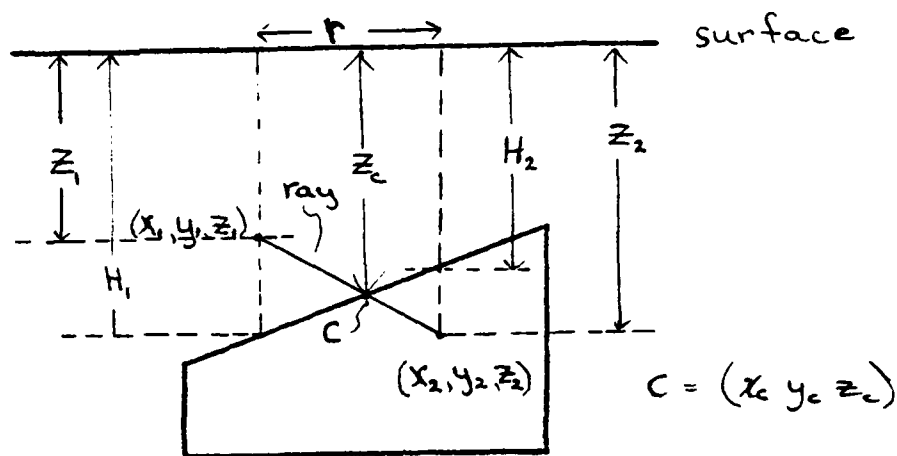
FIGURE III.2. Profile of the bottom cell in the propagation
plane, with a ray contact.

of the projection triangle which contains the ray's horizontal

coordinates x and y.   The subscript parameters are indicated

in Figure III.3.   The following suffixes identify the relative

locations of the vertices of the projection triangle:

    RT - right

    LT - left

    TP - top

    BT - bottom
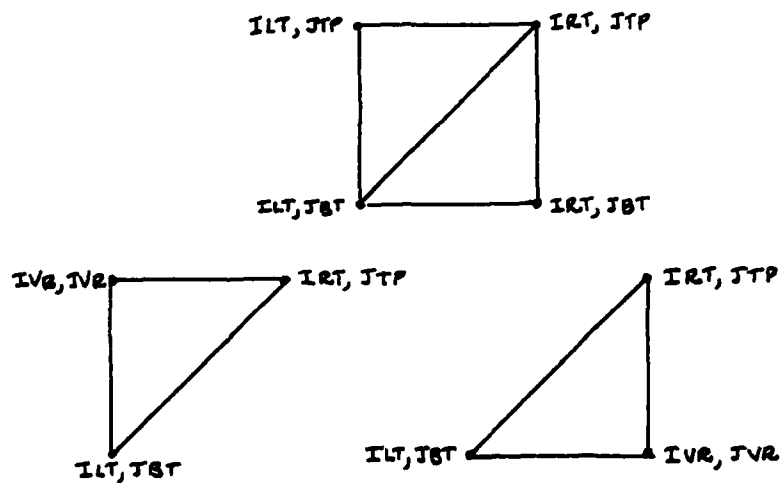
FIGURE III.3.   Projection Triangle Vertices

26

3. Depth Calculation

Subroutine DEEP calculates the depth in kilometers corresponding to the horizontal coordinates x and y. Automatically, IDTSUB is called to identify the projection triangle subscripts. The depth z_c is calculated by solving the equation of a plane defined by the depths at the vertex points (D_1, D_2, and D_v). Figure III.4 illustrates the calculations made based on the orientation of the projection triangle outlining the bottom facet plane. The quantities x_c and y_c represent the horizontal position (in units of DHN) of the depth calculation point relative to the horizontal position of the depth calculation point relative to the horizontal position of the depth value D_1 (the lower left vertex of the projection triangle) as calculated in Secion 1. The depth values D_x and D_y are normalized by the length of the equilateral side 1, in units of DHN. Finally, the depth is solved by

$$z_c = x_c D_x + y_c D_y + D_1$$

B. RAY LOCATION AND DIRECTION PARAMETERS

A three dimensional left hand coordinate system is used (Fig. III.5.), with the positive z axis pointing in the direction of increasing depth and the sea surface level lying in the x-y plane. As the ray is traced through three dimensional space, the program variables which are used to locate the tail of the ray vector are:

DEP1 - depth coordinate

CRE - x coordinate in km

27

$$D_x = \frac{D_2 - D_v}{\ell}$$

$$D_y = \frac{D_v - D_1}{\ell}$$

$$D_x = \frac{D_v - D_1}{\ell}$$

$$D_y = \frac{D_2 - D_v}{\ell}$$
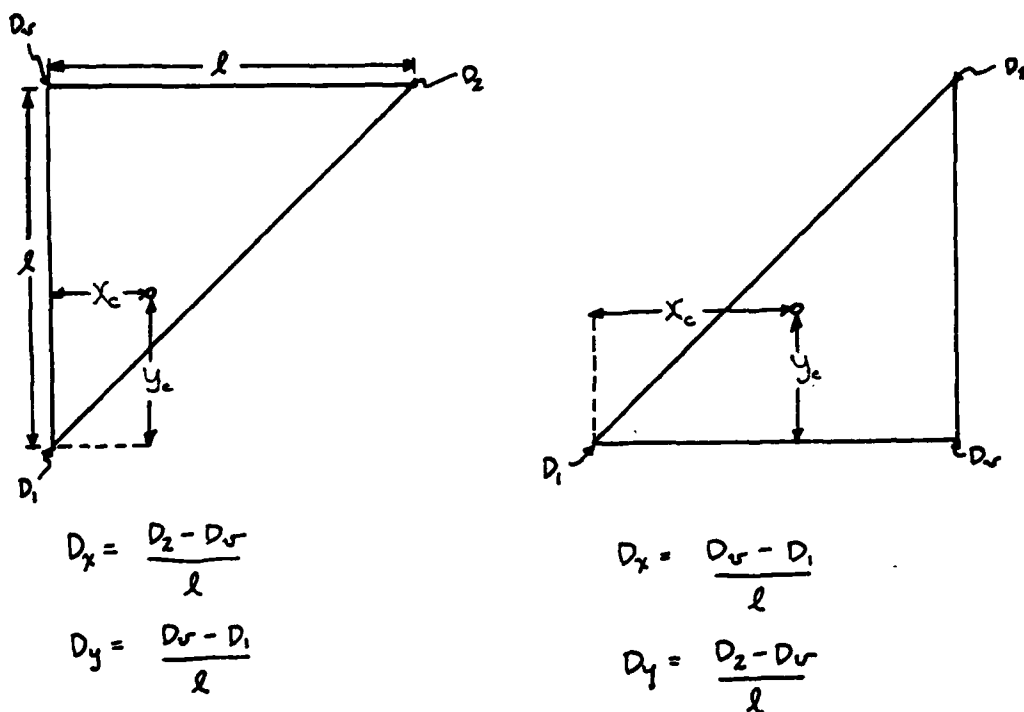
FIGURE III.4.   Depth calculation on bottom facet

CRN - y coordinate in km

CRE and CRN are components of the horizontal range increment DR which forms the propagation plane heading angle PHI with the y-axis.
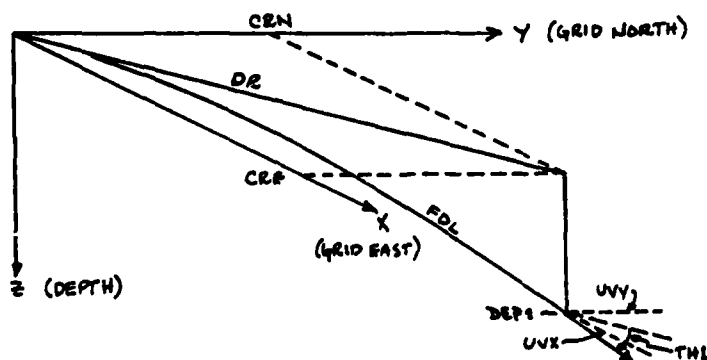
FIGURE III.5.   Ray Location Parameters

Given that the ray vector originates at the point with the above coordinates, the direction of the ray vector is specified by the following angles:

TH1 - ray elevation angle in the plane of propagation measured from the ray vector's projection on the xy plane to the ray vector. A ray pointing toward the ocean bottom will have a positive elevation angle. Theta takes on values between $-\pi/2$ and $+\pi/2$ (output values are in degrees).

PHI - ray (propagation plane) heading angle between the positive y-axis and the projection of the ray vector on the xy plane, measured clockwise from the positive y-axis (grid North). PHI takes on values between $-\pi$ and π.

Initial values for TH1 and PHI are calculated from user input values (in degrees), which are converted to radians for computational purposes. The unit vectors in the horizontal plane are given as UVX = sin (PHI) and UVY = cos (PHI).

C. RAY PATH CALCULATIONS

1. Path Length

ECTRACE confines refraction to the vertical plane alone. It is recognized that while horizontal sound speed gradients exist, their effects are sufficiently small within a well defined water mass to be neglected compared to the bathymetric effect.

However, while the horizontal gradients are likely to be slight, important discontinuities in water characteristics may occur horizontally across oceanic fronts. For this reason,

ECTRACE has been designed to simulate up to three distinct water masses and two associated frontal boundaries.

Within a water mass having uniform characteristics along fixed depths a ray path conveniently defines a vertical plane within which Snell's Law can be written as

$$V = c(z)/\cos\theta(z) \qquad (1)$$

where V is treated as invariant within the water mass (termed the vertex velocity in some texts since it is the speed at the depth at which the ray vector becomes horizontal). The term $c(z)$ is the sound speed at depth z and $\theta(z)$ is the elevation angle at depth z.

Equation (1) uses the simplification that the speed of propagation and the elevation angle are functions of depth only. It is generally possible to define a stratified medium in which each layer's gradient is constant. Although such a model requires a large number of data points to adequately describe a water column of complicated sound speed structure, the piece-wise linear gradient approximation greatly simplifies the ray path descriptions.

Taking advantage of the approximations, we use a simple equation for the sound speed gradient, within a layer,

$$g = dc/dz \qquad (2)$$

Integrating yields

$$z = c/g \qquad (3)$$

where the constant of integration is avoided by arbitrarily specifying the origin of the coordinate axes at the depth where c = 0. By applying Snell's Law we obtain

$$z = \frac{V}{g} \cos\theta \qquad (4)$$

and

$$dz = - \frac{V}{g} \sin\theta\, d\theta \qquad . \qquad (5)$$

Specifying r as the horizontal distance axis of the vertical plane and applying analytical geometry there results

$$dz = \tan\theta\, dr \qquad (6)$$

$$dr = - \frac{V}{g} \cos\theta\, d\theta \qquad (7)$$

and

$$r = - \frac{V}{g} \sin\theta \qquad (8)$$

when measuring r from the vertical (z-axis) to the point at which $\theta = 0$ (horizontal projection).

Squaring (4) and (8) and summing yields

$$r^2 - z^2 = (\frac{V}{g})^2 \qquad (9)$$

This is the equation of a circle of radius (V/g) and whose center is at the origin of the coordinate system just described. The equation

$$R = - \frac{V}{g} \qquad (10)$$

describes the radius of curvature of a ray path in a constant

31

sound speed gradient. The minus sign has been chosen to allow
ρ to be positive when the refracted ray is increasing its ele-
vation angle.

The coordinate system and computer-approximated ray
path segments are shown in Fig. III.6.
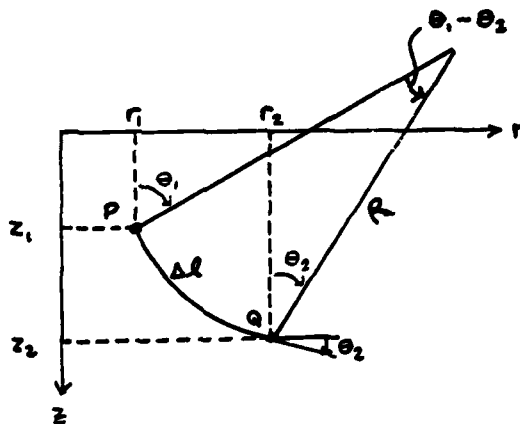
FIGURE III.6. Ray Path Geometry

It can be shown from Fig. III.6. that the ray segment
path length between P and Q is

$$\Delta \ell = \rho \Delta \theta \qquad (11)$$

where

$$\Delta \theta = \theta_2 - \theta_1 \qquad (12)$$

Since the initial elevation angle of the ray is presumed known,
successive values of θ are found iteratively:

$$\theta_2 = \theta_1 + \frac{\Delta \ell}{\rho} \qquad (13)$$

Simple geometry also allows for solution of the depth change,

$$\Delta z = z_2 - z_1 = \rho(\cos\theta_1 - \cos\theta_2) \quad , \qquad (14)$$

and likewise the increase in horizontal range,

$$\Delta r = r_2 - r_1 = \rho(\sin\theta_2 - \sin\theta_1) \quad . \qquad (15)$$

Note that in the Fig. III.6., $\Delta\ell$, Δz, and Δr must all be positive, so the sign of ρ must agree with that of $\Delta\theta$. Equation (10), which gives ρ the opposite sign of the gradient, satisfied this requirement. Only Δz is allowed to become negative, as when a ray decreases its depth.

2. Travel Time

From the diagram (Fig. III.7.) the sound speed relationship becomes

$$c = \frac{df}{dt} \qquad (16)$$

$$dt = \frac{dz}{c\sin\theta} \qquad (17)$$
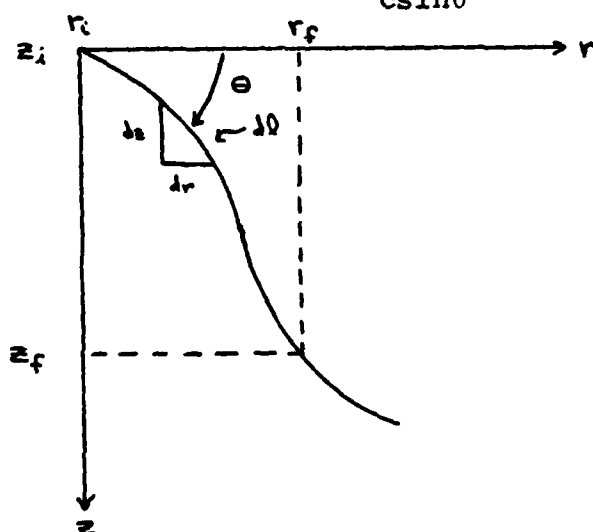
FIGURE III.7. Ray path and sound speed relationship.

The ray invariant

$$a = \frac{1}{V} = \frac{\cos\theta}{c} \qquad (18)$$

and the trigonometric identify $\sin\theta = \sqrt{1 - \cos^2\theta}$ combine and give dt as a function of depth

$$dt = \frac{dz}{c(t) \sqrt{1 - [ac(z)]^2}} \qquad (19)$$

The time-of-flight of the acoustic ray along the path is found by integrating from the initial point to the final point:

$$t_f - t_i = \int_{z_i}^{z_f} \frac{dz}{c(z) \sqrt{1 - [ac(z)]^2}} \qquad (20)$$

Before integrating it is assumed that the gradient is constant along the segment. More generally, ECTRACE assumes a constant gradient in the layer bounded by z_1 and z_2.
Then

$$c(z) = c(z_1) + g(z_2 - z_1) \qquad (21)$$

where $z_1 \le z \le z_2$.
The integral may now be evaluated by introducing a new variable $W = \frac{c(z)}{g}$ such that

$$W = z - z_1 + \frac{c(z)}{g} \qquad (22)$$

Then $dW = dz$ and $C(W) = gW$. Therefore,

$$t_f - t_i = \int_{W_i}^{W_f} \frac{dw}{gW \sqrt{1 - a^2 g^2 w^2}}$$

34

$$= \frac{1}{g} \ln \left\{ \frac{W_f [1 + \sqrt{1 - a^2 g^2 Wc^2}]}{W_i [1 + \sqrt{1 - a^2 g^2 Wf^2}]} \right\} \qquad (23)$$

$$= \frac{1}{g} \ln \left[\frac{W_f (1 + \sin\theta_i)}{W_i (1 + \sin\theta_f)} \right] . \qquad (24)$$

Values for W are restricted to being in the same layer and on the same side of a turning point in the ray path.

D. RAY TRACING PROCEDURE

ECTRACE begins tracing a ray from its given initial coordinates CRE, CRN, and DEP1. First the water mass is identified to determine the local gradient and therefore the ray invariant and radius of curvature. The ray path is constructed in the form of arc segments using an increment DTH calculated from a chosen arc length DL. Then the new θ value is calculated and Equations (14) and (15) are used to determine the ray segment's depth increment and horizontal range increment. The horizontal position is updated using the horizontal unit direction vectors (UVX and UVY) calculated from PHI, the initial heading of the ray's plane of propagation.

The interim update 3-D coordinates of the ray head may now be stated in FORTRAN as

$$CEE = CRE + DR * UVX \qquad (25)$$

$$CNN = CRN + DR * UVY \qquad (26)$$

$$DEP2 = DEP1 \text{ and } DZ \qquad (27)$$

where DR represents the horizontal range increment Δr and DZ is the depth increment Δz.

E. RAY CONSTRAINT CONDITIONS

Each time the ray path is incremented, certain constraint conditions must be checked. These constraints are bottom reflections and the sound channel limits.

1. Bottom Reflections

After the interim horizontal coordinates CEE and CNN are calculated, they are passed to DEEP to find the bottom depth WH2 at the ray head. If DEP2 is greater than or equal to WH2, a bottom bounce has occurred. In this case the values of DEP2, CEE, and CNN are adjusted by linear interpolation and a point of bottom contact is established. Subroutine BOANG is then called to calculate new values of TH1, PHI, UVY, and the ray grazing angle GRAZ.
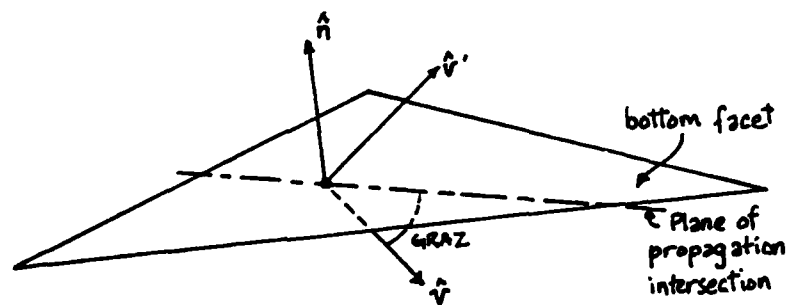
FIGURE III.8. Bottom Bounce Reflection Angles.

The basic calculations of BOANG involve the vectors:

$\hat{v}$ - unit ray vector before bounce

$\hat{n}$ - unit vector normal to triangle facet at the bounce point

$\hat{v}'$- unit ray vector after bounce

The components of $\hat{v}$ are calculated from TH1 and PHI. The components of $\hat{n}$ are calculated from the equation of the plane of the bottom facet. The new vector $\hat{v}$ is then calculated from

the vector equation,

$$\hat{v}' = \hat{v} - 2 \, (\hat{v} \cdot \hat{n}) \, \hat{n} \quad . \tag{28}$$

New values of TH1, PHI, UVX and UVY are then calculated from the compnents of $\hat{v}'$. Finally, the grazing angle GRAZ comes from

$$\sin(GRAZ) = -(\hat{v} \cdot \hat{n}) \quad . \tag{29}$$

Reflection from a sloping bottom plane facet, as opposed to a surface reflection, forces a reevaluation of the Snell's Law constant V, and therefore the radius of curvature R. The invariant V remains in effect until the next bottom bounce occurs or a new water mass is entered. Thus, after summoning BOANG, the ray tracing continues with the new ray parameters. The sound channel limits CHS and CHD described below are also recalculated at this point since they also depend on V, and GRAZ is passed to function FBLOSS to calculate a bottom loss value.

2. Sound Channel Limits

The Snell's Law equation (1) says that the ray trace is constrained to depths where the speed of sound does not exceed V. A given ray path may be constrained in the sense that it is purely refracted (R), refracted or reflected (RSR or RBR) or purely reflected (BRSR). The determining factors are V and the local sound speed by profile. V is calculated from the sound speed at the initial depth and the initial elevation angle, and is recalculated after a bottom bounce or upon entering a new water mass.

37

CHS and CHD are the refractive sound channel limits
at the shallow and deep extents respectively. Since the sound
speed profile is approximated by piecewise continuous linear
gradients, the water column is treated as a column of indexed
layers, each characterized by upper and lower (shallow and
deep) depth boundaries and a constant sound speed gradient.
The limiting depths CHS and CHD are found by scanning the
water mass layers both above and below the current ray loca-
tion until a boundary sound speed value is found which exceeds
V. The limiting depth is then calculated from

$$z_{lim} = z + (V - c)/g \qquad\qquad (30)$$

where z and c are evaluated at the top of the layer and g is
the gradient of the limiting layer.

For cases where g = 0 or where V is not exceeded by
any sound speed in the local profile, CHS and CHD are set to
the physical limits of the water column.

F. RAY PROPAGATION TERMINATION CONDITIONS

Tests are made during each ray segment iteration for events
which terminate the ray trace. These events are:

- Escape from grid - when the ray departs the finite x and
  y constraints of the working depth array (ZB).
- Depth boundaries - when the depth detected at the ray is
  less than ten wavelengths or greater than 10 km.
- Bottom loss - when the attenuation due to bottom reflec-
  tion exceeds a specified input amount in dB.

- Error at bottom contact – when defective program code
  or anomalous depth data prevent the interpolative rou-
  tine CONTAC from recovering the bottom depth within a
  reasonable number (50) of iterative attempts.
- Trapping – when it becomes apparent that the ray will
  not contact the bottom, rendering further three-dimen-
  sional tracing unnecessary.
- Bottom bounce gate – when the total number of bottom
  reflections exceeds a user specified amount.

The cause of ray termination is specified in the printed
ray history, giving the ray parameters at the time of
termination.

G. IMPROVEMENTS MADE OVER THE ORIGINAL NRL VERSION

The ray tracing algorithm originally existed as a program
called ABOUNC at the Naval Research Laboratory. ECTRACE dif-
fers significantly from ABOUNC in the following areas:

1. Ray Path Development

ABOUNC calculates ray parameters through right tri-
angle solutions and Snell's Law, but develops the ray path
along fixed depth. Since this prevents rays from becoming
horizontal, a reversal is forced when the magnitude of the
elevation angle goes below a gate value. In contrast, ECTRACE
develops the ray path along arc segments and radii of curva-
ture, fixing only the maximum length of the segments while
interrupting the increment as necessary when a segment reaches
a reversal point or layer boundary, or to keep the increment
within a maximum amount.

39

2. Computer Word Storage

ABOUNC floating point variables are nearly all FORTRAN double precision. The largest single burden on computer core is the depth matrix, read from a magnetic tape in card image format, each card record containing eight items (fields) ten bytes long. This format permitted a depth range of ±9999 km with resolution of ±1 cm. By reducing the depth range to ±999 km and resolution to ±10 cm, two unnecessary bytes are removed from each field, enabling a fit of ten items per record and a permanent storage reduction of twenty percent. In addition, ECTRACE reads the matrix into a single precision array to reduce the core requirements for the data by fifty percent.

3. Grid File Manipulation

ABOUNC reads the contents of the entire depth matrix data file into array whose dimensions must be exactly equal to those of the matrix. This requirement is not inconvenient on a large computer with FORTRAN dialect which permits objects time dimensioning. ECTRACE is written in FORTRAN G for an IBM Series/360 Model 67 machine whose limitations required some fundamental modifications of the program and its job control.

To maintain flexibility and reduce run time, ECTRACE was reduced to a package of subprograms which have been pre-compiled and stored on a permanent device in load module form. The user must supply a calling program which serves primarily to establish the dimensions of the depth array. The user's selection of array dimensions are made on the basis of core economy and area of interest rather than the dimensions of the

input matrix. ECTRACE, through its internal subroutine GRDSCT, extracts a rectangular working subregion from any part of the input matrix. Trial runs of ECTRACE used a working array of dimensions 151 by 151 representing a sector of 150 km on a side with a core requirement of less than 250K, while the input matrix was 369 by 443. Without GRDSCT, ECTRACE would have required an allocation of 800K bytes using the same input file.

4. Treatment of Small Elevation Angles

As stated previously, ABOUNC forced a reversal of the ray as the elevation angle of the leading segment approached a gate value close to zero. Since the ECTRACE tracing scheme permits a full range of elevation angles, early trial runs revealed an inherent error phenomenon as θ approached zero in a steep gradient. Specifically, a $\Delta\theta$ on the order of 3 degrees caused by a small radius of curvature or along arc segment traced a ray to its vertex depth in error as much as 50 meters from the Snell's Law prediction. Since the depth change was calculated from the difference in elevation angle cosines, the error was found to arise from the machine inability to retain difference precision for any pair of numbers close to integer values (round off error). Since the difference rather than the actual cosine values is needed for this calculation, the small angle series approximation $\cos \quad 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} - \ldots$ was used to formulate the equation

$$\cos\theta_1 - \cos\theta_2 \cong \tfrac{1}{2}(\theta_2^2 - \theta_1^2) - \tfrac{1}{24}(\theta_2^4 - \theta_1^4). \qquad (31)$$

When restricted to small angles, this approximation yields

41

greater accuracy than the straightforward machine calculation of the left hand side, since the precision retained by the registers is very high.

5. Travel Time

ABOUNC calculates the travel time of an acoustic signal along a ray segment by dividing the linearly-approximated path length by the mean sound speed. ECTRACE accounts for path curvature and depth dependence of the sound speed through integral evaluation, discussed in detail in section C.2.

6. Track Events

Layer transitions, reflections, and turnarounds are events which may require a change in ray parameters. ABOUNC tests for these events in algorithmic sequence, makes the necessary changes in response to the first event detected and continues the trace from the current position of the ray vector head. ECTRACE retraces a ray segment to the event depth and tests for all other events before beginning the next segment from the transition point. This logic eliminates the possibility of a long ray segment crossing more than one event depth with only one event detected.

IV. USER INSTRUCTIONS (ECTRACE)

A single step ECTRACE job traces a bundle of rays from a single source in the form of ray fans. Each ray fan is a bundle of rays of common initial elevation angle. All ray fans are bounded between to initial heading values measured from grid North (grid North equals true North at the grid center).

During the following discussion the reader may wish to refer to Fortran listing in Appendix M. The user instructions for the other programs (GENBOT, G3DP, SYNGEN, GRDCHK, and ECCOM) are explained in their individual appendices (User Instructions and Output Description).

A. JOB CONTROL CARDS (JCL)

The following JCL cards and parameters are required for an ECTRACE run:

- JOB card - The CPU time parameter should allow 60 secconds for every seven rays to be traced.
- EXEC card - This card must specify the FORTCLGW procedure (Appendix O). 250K bytes of core is required for the standard 22801 (151 by 151) element depth matrix portion plus 4K for every additional 1000 elements. The computer systems job class definitions should be considered before deciding upon the size of the working matrix and the number of rays. For example,

      EXEC FORTCLGW,REGION.GO=250K

  is a class C job at NPS when limited to five minutes of

time, which is adequate for tracing 40 rays.

- Calling program - Following the

    //FORT.SYSIN DD *

card, the calling program must contain cards

    COMMON/DIMS/KMX,KMY and

    CALL TRACER (ZB) .

ZB is a REAL * 4 array dimensioned exactly by KMX,KMY, and

these values (KMX,KMY) are initialized to the values

of the DIMENSION statement using assignment statements,

as in the following example.

```
DIMENSION ZB(151,151)

COMMON /DIMS/ KMX,KMY
.
.
KMX=151

KMY=151
.
.
CALL TRACER(ZB)
.
.
STOP

END
```

KMX and KMY are INTEGER * 4 and should be less than or

equal to the dimensions of the data matrix from which

the depth values will be taken.

- LINK cards - These cards point to the ECTRACE load

module. They must include all members of the module

for which the user does not supply a substitute. An

example which includes all members is:

```
//LINK.USDD UNIT=3330,VOL=SER=DISK01,DSN=1270.ECTRACE,

//   DISP=SHR,LABEL=(,,,IN)

//LINK.SYSIN DD *
```

```
        INCLUDE USDD(TRACER,GRDSCT,IDPROF,NUPROF,CHNLIM)

        INCLUDE USDD(BGNPLT,SSPPLT,BDTPLT,RNGPLT,T2DPLT,ENDPLT)

        INCLUDE USDD(IDTSUB,DEEP,CONTAC,BOANG,ANGPRT,FBLOSS)

        ENTRY MAIN

        /*
```

• GO cards - ECTRACE uses data set reference number one

for input of the depth matrix. For example, to point

to a depth matrix residing in a data set called S9999.GRID

on DISK03, the following DD statement is used:

```
    //GO.FT01F001 DD UNIT=330,VOL=SER=DISK03,DSN=S9999.

        GRID,DISP=SHR,LABEL=(,,,IN)
```

If a depth matrix from an outside source is used, it

must be in the format specified in the Output Grid Data

Format section of Appendix A. If a punched card output

is not desired to make an ECCOM run, include the card

```
    //GO.FT07F001 DD DUMMY                        .
```

ahead of the GO.FT01F001 DD statement shown above.

B. DATA CARDS

    Following the

```
        //GO.SYSIN DD *
```

card, the user selected data must be provided in the following

format. FORTRAN field descriptors are in parentheses. The

FORTRAN variable names are listed for reference.

    • Card 1 (20A4)

        OTIT - title of the run

    • Card 2 (9S8.2)

        X0 - source position for x coordinate (kilometers).

YO - source position y coordinate (kilometers).

ZO - source depth (meters). A source depth which exceeds the water depth will be decreased to one meter above the bottom. Source coordinates in a nonpropagating location will also be adjusted.

XREC - receiver position x coordinate (kilometers).

YREC - receiver position y coordinate (kilometers).

ZREC - receiver depth (meters). The receiver may be positioned at the source if calcuations are not desired. Excessive depth is automatically corrected.

FREQ - frequency (Hz). A ray is terminated when the water depth becomes less than ten wavelengths during propagation.

FDL - (optional) segment length (meters). The parameter affects the smoothness of the vertical plot. Smaller values increase smoothness and run time. Large values create the risk of missing a bottom peak. The default value is one wavelength.

- Card 3 (6F8.2)

XMIN - initial x value of horizontal plot (kilometers).

YMIN - initial y value of horizontal plot (kilometers).

The above values are used for drawing the axis of the horizontal plot and for determining the section of the input matrix to be read into a working depth matrix (ZB). The values for KMX and KMY, recorded cell resolution (DHN), and the limits of the input matrix ultimately

determine the axis ranges. The user should be familiar with these parameters when choosing the origin of the plot.

ZST - initial depth value for vertical plot (kilometers)

ZEND - final depth value for vertical plot (kilometers)

ARST - initial range value for vertical plot (kilometers)

AREND- final range value for vertical plot (kilometers)

The above values are used to draw the axes of the vertical plot. The initial values may be negative. The program adjusts the final values if necessary for plot esthetics.

- Card 4 (6F8.2)

HDST - heading of the first ray in each ray fan (degrees)

HDEND - heading of the last ray in each ray fan (degrees)

ELST - elevation angle of the first ray fan (degrees)

ELEND - elevation angle of the last ray fan (degrees)

Headings are measured from grid north. Elevations are measured from the horizontal plane, positive downward (toward increasing depth).

HORLIM - maximum ducting or channeling range (kilometers). This is a gate value, like BLMAX CARD 2, which terminates the trace of a ray when exceeded. HORLIM serves to screen out those rays which do not appear to contact the bottom.

SMNGL - (optional) small angle value for alternative cosine difference calculation (degrees). A value of less than three degrees is recommended. See the discussion on "small angle approximations"

in Chapter III. The default value of two
degrees is set when left blank.

- Card 5 (6I5)

    NHD — number of rays per ray fan (≥ 1)

    NEL — number of ray fans per bundle(≥ 1)

    NSTOP — maximum number of bottom bounces per ray (≥ 1)

The program restricts NHD to 30 rays, with a maximum
total number of 100 rays produced, each with a maximum
number of 50 bottom bounces or 200 reversals.

    INCPLT — number of ray segments defining one plot seg-
    ment (≥ 1). With sufficiently small ray seg-
    ments it is not necessary to use each iteration
    for plot definition to achieve a smooth plot,
    so this parameter is provided to enable the
    user to coordinate FDL and INCPLT for run
    time economy.

    NLEN — length of the longest horizontal plot axis
    (≤ 19 inches). This value controls the physi-
    cal size of the plot. A value less than four
    will supress the plot (resulting in an ABEND
    code in the HASP log), but it will not affect
    the calculations or printed and punched out-
    put. A value greater than 15 requires addi-
    tional JCL for strip plotting, since the
    total plot width is NLEN plus six inches.
    Refer to Ref. 7 for strip plotting.

NSSP - number of sound speed profiles (water masses
provided as data (1 ≤ NSSP ≤ 3).

- Card 6 (A4,I3)

    OMAS (1) - name of first water mass

    NVAL (1) - number of (depth, sound speed) pairs defining
    the sound speed profile (1 ≤ NVAL ≤ 50).

- Card 7 - (through card 16 if necessary) (10F8.2)

    These are the data pairs (depth, sound speed), in meters
    an m/sec., from the sea surface to the last data point
    in the profile. The gradients will be calculated linearly
    between these points, with the final gradient below the
    last point set to 0.01668 m/sec. by the program.

- Next card (4F8.2)

    If the previous data is to be followed by the data for
    another water mass, this card must be included to define
    the frontal boundary between the two. The order in
    which the water masses appear as inptut determines their
    positions relative to the fronts. Cards which describe
    the next water mass, follow the frontal definition card
    and are formatted as described cards 6 and beyond. The
    last data card is the last string of (depth, sound speed)
    pairs of the last water mass; refer to Appendix M (sec-
    ond page) for an example of three water masses. The
    following variables define the boundary between each
    water mass:

    FENTX (1) - x coordinate of front beginning point (km)

    FENTY (1) - y coordinate of front beginning point (km)

FENTX (l) - x coordinate of front end point (km)

FEXTY (l) - y coordinate of front end point (km)

The examples shown in Fig. IV.1 identify the water
mass boundaries for various input axis values in the
form (FENTX, FENTY), (FENTX, FEXTY) for each water mass
boundary. The first example a. represents the data
from an original ECTRACE run, refer to the second page
of Appendix M. Exambles b. and d. show the priority
of the water masses. The first water mass defined,
Northwestern Mohns Ridge (NWMR), dominates the second,
Southeastern Mohns Ridge (SEMR), and the third, isove-
locity (ISOV), where they overlap.

a.
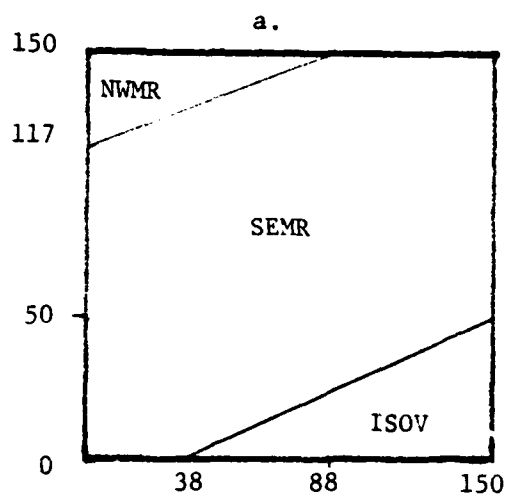
NWMR
  (0,117) (88,150)
SEMR
  (38,0) (150,50)
ISOV

b.
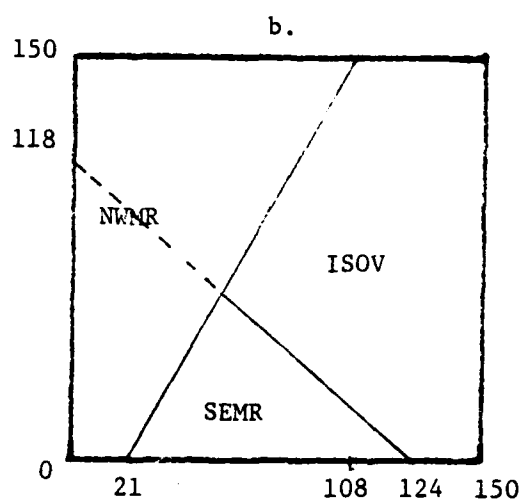
NWMR
  (21,0) (108,150)
SEMR
  (0,118) (124,0)
ISOV

c.
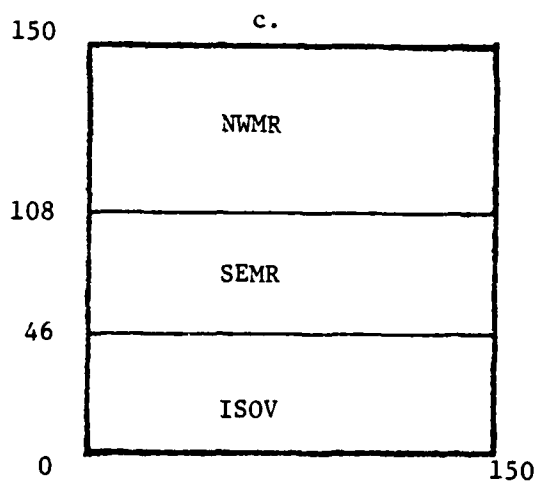
NWMR
  (0,108) (150,108)
SEMR
  (0,46) (150,46)
ISOV

d.
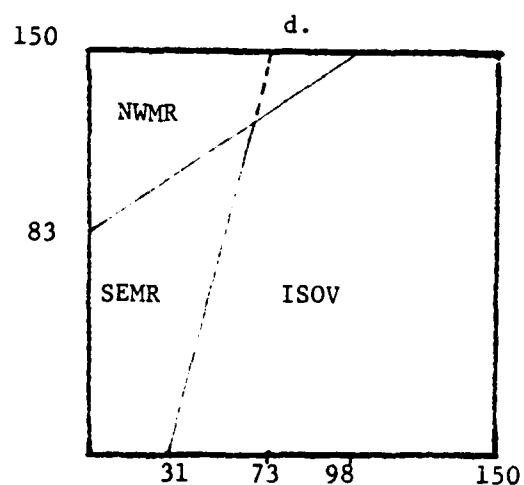
NWMR
  (0,83) (98,150)
SEMR
  (31,0) (73,150)
ISOV

FIGURE IV.1.   Examples of water mass boundaries for various values of
                (FENTX,FENTY) and (FEXTX,FEXTY).

51

V. INTERPRETATION OF ECTRACE OUTPUT

The following discussion refers to an ECTRACE run titled "Trial 67***Synthetic Basin***" (Figure 5), which is a trace conducted in a synthetic parabolic basin (Figure 6). A portion of the printed output is shown in Tables II - V. In this ECTRACE run, eight ray fans were selected to be projected between -30 and +30 degrees elevation. Each ray fan was composed of only one ray, thus the initial heading for each ray was 010 degrees.

A. PRINTED OUTPUT

Refer to Table II through V (Appendix G) as examples of the following discussion.

1. Parameter Tableau   (Table II)

After processing and adjusting the input parameters just described, ECTRACE prints a final tableau of these parameters plus some additional parameters calculated from the original input such as layer gradients, the total number of rays to be traced, and the spacing between rays. In addition, the results of the depth matrix-grid load is summarized, revealing data from the header record of the grid file.

2. Ray History Tableau   (Table III)

During the trace of each individual ray, this tableau prints a line of ray parameters at each end point and reversal. The following is a description of one line of parameters.

• TYPE identifies the nature of the reversal, and is

abbreviated INIT for initial position, BOTT for bottom
bounce, SURF for surface reflection, OVER for turnover,
and STOP for terminated position.

- X, Y, and DEPTH are the coordinates in kilometers of the
  reversal and end point.
- GRID HEADING is the ray heading measured clockwise from
  grid North at the completion of the reversal.
- ELEV is the ray's elevation angle measured downward
  (positive toward increasing depth) from the horizontal
  zero degree reference).
- BRNG TO REC is the bearing from the reversal point to
  the receiver position.
- DIST TO REC is the slant range distance from the reversal
  point to the receiver.
- WTR MASS is the water mass identifier indicating the
  sound speed profile applied in the present calculations.
- LYR is the constant gradient layer number corresponding
  to the order input as profile data (i.e., layer 1 is the
  surface layer and indicates that the surface gradient
  applies.
- DEPTH LIMITS are the physical and refractive sound chan-
  nel limits of the ray, determined by its invariant. If
  MIN is 0.0, an upward traveling ray will reflect off
  the surface before reaching its upper vertex (turnover)
  depth. If the MAX depth is greater than or equal to 10km,
  a line of asterisks is printed since the ray could never
  attain its turnunder depth before reaching the bottom.

Otherwise, these limits are the ray's vertex depths, and if the reversal type is OVER or UNDR, the appropriate limit should exactly match the depth value on the same line.

- C is the sound speed at the reversal point. This is the ray invariant used for tracing.

- GRAZ ANGLE is the angle in degrees between the ray vector and the line of intersection of the bottom facet plane with the vertical plane. This is the angle used to calcualte bottom loss and the new ray direction vector.

- BTM ANGLE is the angle between the bottom facet plane normal and the vertical (z-axis), in degrees.

- BTM LOSS is the accumulated intensity loss in dB after the bottom reflection. When this value equals or exceeds BLMAX, the ray's trace will terminate.

- LPS is the number of steps required by the CONTAC subroutine to locate the point of bottom contact. A long ray segment length may require more steps than a shorter one. A maximum number of 50 steps is permitted before the bottom facet is assumed anomalous and the ray's trace is terminated.

- Prior to the final line of an individual ray's trace history, is a summary of plot definition parameters plus data on the ray's CPA to the receiver, and reason for ray trace termination. The CPA calcualtions are accurate to within one ray segment length.

3. Ray Fan Summary Tableau (Table IV)

At the completion of the trace of one complete ray fan, a table is printed listing each ray's elevation angle and depth in meters before each bottom bounce. All zero values at the bottom of each ray's column indicate blank data.

4. CPA Summary Tableau (Table V)

When all ray traces have been completed, a table is printed listing each ray's heading, time, and distance at CPA to the receiver.

B. PLOTTED OUTPUT

ECTRACE produces a single montage of up to five separate plots (Fig. 13, the total dimensions of which depend upon the user's selection for the maximum size in inches of the longest axis of the horizontal plot (NLEN). The following table lists some sample values for NLEN and the resulting dimensions of the complete plot.

NLEN	Maximum (in.) Height	Maximum Width	Remarks
5	7	8.33	minimum
10	14	16.67	
15	21	25	maximum size without strip plotting [5]
19	26.6	31.67	maximum allowable

Components of the ECTRACE plot follow.

1. Sound Speed Profiles (Figure 13)

A plot of sound speed versus depth is made for each water mass defined.

2. Vertical Plot (Figs. 15 and 16)

Most two-dimensional ray trace programs produce plots of ray paths on graphs of depth versus range. The vertical plot is similar except that all ray paths are projected from three dimensions onto the vertical plane. The plane is oriented along the mean ray fan heading and includes its profile of the sea bed. In ECTRACE trial 65B (Figure 13) the ray fans each contain two rays whose initial headings vary from 360 to 020 degrees as indicated by 10.000 +/- 10.000 on the plot. The bathymetry profile is made along the heading of 010 degrees. On a plot, the heading value (HEADINGS) always indicates the direction of the bathymetry slice. The (+/-) value signifies the initial heading of the first and last rays and that additional rays are evenly spaced in the interval.

Since the rays are actually traced in three dimensions, bottom reflections occurring outside the vertical plot plane may appear to be taking place above or below the bottom surface instead of at the boundary (Fig. 15). Only those bottom contact points which occur in the projection plane must lie on the seabed profile.

The symbols listed on the plot (Fig. 5) are used to identify bottom contact locations in the vertical and horizontal plot, and are unique for each ray fan except that up to thirteen symbols are used before repetition, as shown in Table VI.

3. Horizontal Plot (Fig. 17)

This plot is the most important plot product of ECTRACE and is the plot of the ray paths on a horizontal plane

(topview). This plot displays the X-Y positions of each bottom contact point using the ray's (ray fan) identification symbol and connects them with straight lines, revealing the horizontal deflection effect. Each ray's plot begins at the source, ends up at the last bottom contact point in the grid and is labeled with the ray number (Fig. 17). The ray number can be used to locate its printed and punched output history.

The axes are labeled with respect to the origin of the input matrix. The geographic center position and coordinates of the input matrix, if included in the bounds of the working matrix (FORTRAN array ZB), is also plotted for reference. It is important to note that this plot, without the bottom contact symbols, may combine the results of other ECTRACE runs by combining the optional punched card outputs as input to ECCOM (Figs. 7 and 18).

C. PUNCHED OUTPUT

The punched card output for a single ECTRACE job consists of the following cards:

CARD	DATA CONTENTS
1	The runtitle
2-3	The grid title
4	Grid reference coordinates and scale factor
5	Initial elevation, initial heading, number of points (BB) and ray number of the first ray
6+	X-Y coordinate pairs of plot points of the first ray
(Data for additional rays is repeated as 5 and 6+ above)	
Last	9999., to flag the end of data

These cards may be stacked as a member with those of
several ECTRACE jobs as input to a single ECCOM job. For
ECCOM to be meaningful, cards 2 through 4 of each stack mem-
ber should be identical, and the user should verify this
himself.

Punched card output may be quite voluminous and should
not be produced unless required. The following card, inserted
before the first GO card (JCL), will suppress the punched
output:

    //GO.FT07F001 DD DUMMY   .

VI. SUMMARY

A. POTENTIAL APPLICATIONS

ECTRACE, in its present form, has many potential
applications:

- The investigation of asympotic deflection angles in
  ocean areas where bathymetry data are well-documented,
  allows optimum hydrophone positioning. For example,
  Ref. 6 cites cases where seemingly attractive receiver
  positions would be affected by shadowing in the hori-
  zontal plane.

- Adaptation of ECTRACE ray tracing methods could be
  incorporated into existing programs which presently
  rely upon an assumption of radial symmetry of ray
  propagation about a source or receiver (no horizontal
  deflection).

- The horizontal deflection effect may lend additional
  insight into the study of travel times and intensities
  of acoustic signals, enabling more accurate source
  fixing information.

B. IMPROVEMENT

There are many areas of improvement which would increase
the utility of ECTRACE and its supporting computer programs
for the above applications:

- A better routine to convert bathymetry contour data
  into a matrix of depth values is essential for modeling

regions for which only contour data exists. Most known interpolation algorithms perform best on data which are randomly or regularly spaced rather than in the form of contours. For example RAIN2 [5] would probably perform better with the data used to make the contours than it does with the contours themselves. While the contour data are useful for cartographic and geologic purposes, bathymetry data in a less processed form are significantly better sources for constructing computer models of the sea bed.

- ECTRACE in its present form makes no calculations of intensities or their losses due to spreading, attenuation, or scattering, thus preventing its use as a propagation loss model.

- An improved bottom loss function is recommended over the supplied FORTRAN function subprogram FBLOSS. The user may perform this substitution easily by deleting FBLOSS from the INCLUDE step in the LINK statement of the ECTRACE job control and supplying another function subprogram (added at the end of the calling program) that accepts a double-precision argument for grazing angle and is also called FBLOSS.

- The assumption of specular reflection neglects the phenomenon of transmission into the sediment and its effect on the location of the reflected ray due to sub-bottom boundaries. It would be interesting to incorporate in an improved version of ECTRACE.

APPENDIX A
GENBOT USER INSTRUCTIONS AND OUTPUT DESCRIPTION

GENBOT is used in the construction of a depth grid file from bathymetry contour files. In this discussion, the term "grid" refers to a matrix of depth values whose row and column dimensions represent data point spacing in terms of a desired resolution parameter DHN.

1. Selection of Bathymetry Grid Regions

GENBOT will create a bathymetry grid file for any region of the geoid for which contour data is supplied in the format stated in section A-4. The user must select the latitude and longitude of the center of the region of interest, a radius in kilometers and the cell resolution in kilometers. GENBOT will attempt to create a square matrix whose semi-axes represent the selected radius and whose elements represent depth values in kilometers.

A 30-km data margin, outside the calculated axis limits, is required to begin interpolation. Should this requirement not be met, GENBOT will symmetrically reduce the length of one or both axes until the resulting rectangular matrix plus margins with the desired center position intact is totally encompassed within the data region. The center coordinates, area radius, and cell resolution must be selected so as to avoid generating a grid which is impracticably small. In particular, a generated grid may not overlap a pole, and one with a pole on a margin boundary must be supplied data from eighteen contour files to cover 180 degrees of longitude.

Contour data currently available in usable format cover 40 degrees of longitude of the North Atlantic Ocean, specifically the Norwegian and Greenland Seas, from longitudes 20W to 20E and latitudes 60N to 90N (Fig. 8). The Naval Research Laboratory, Acoustics Division, Arctic Section,

has additional files which cover the entire North Atlantic land and sea regions which may be utilized by GENBOT after modification. However, GENBOT need not be the exclusive means of grid instruction. The only requirement is that the grid exists as a file in the format specified. The user may wish to draw on other souces of bathymetry data, such as SYNBAPS [8], and tailor other available resources to construct a usable grid.

2. Grid Construction Techniques

Once supplied with the user's selection for the grid title, center coordinates, cell resolution and area radius (TTL, CNLAT, CNLON, DHN, ARAD), GENBOT begins by defining a north-oriented square region whose boundaries are at a distance ARAD from the center. The subscripts of the two dimension matrix depicting this region will represent equal units of DHN spacing from the origin along the x and y axes of the square using a cartesian coordinate system. Thus, while all distances calculated during grid generations are geodetic distances from the center, point positions are made relative to the grid origin or southwest corner. This technique results in a plane projection which is distortion free along straight tracks which pass through the center. Subroutine CORNER then calculates and prints a table of the initial cartesian and geographic coordinates of the grid corners. The cartesian coordinates will always range from (0,0) to their maxium values, initially double the value of ARAD. The geographic coordinates should reflect some meridian convergency. This table will be repeated for the final grid, showing the adjustments made, if any, to meet the data requirements.

Many geographic coordinate pairs in an input file may be well outside the area relevant to the grid, so GENBOT calculates gate values called

62

SLAT, ELAT, SLON, and ELON for data discrimination. SLAT and ELAT are always one degree beyond the minimum and maximum latitude values calculated by CORNER, but SLON and ELON may extend beyond the minimum and maximum longitudes by 1, 10, or 20 degrees, depending upon the maximum latitude.

The grid region and its data environs recognized by GENBOT at this stage is shown in Fig. A.1. All points within these environs are subject for consideration during interpolation of depth values in the grid.
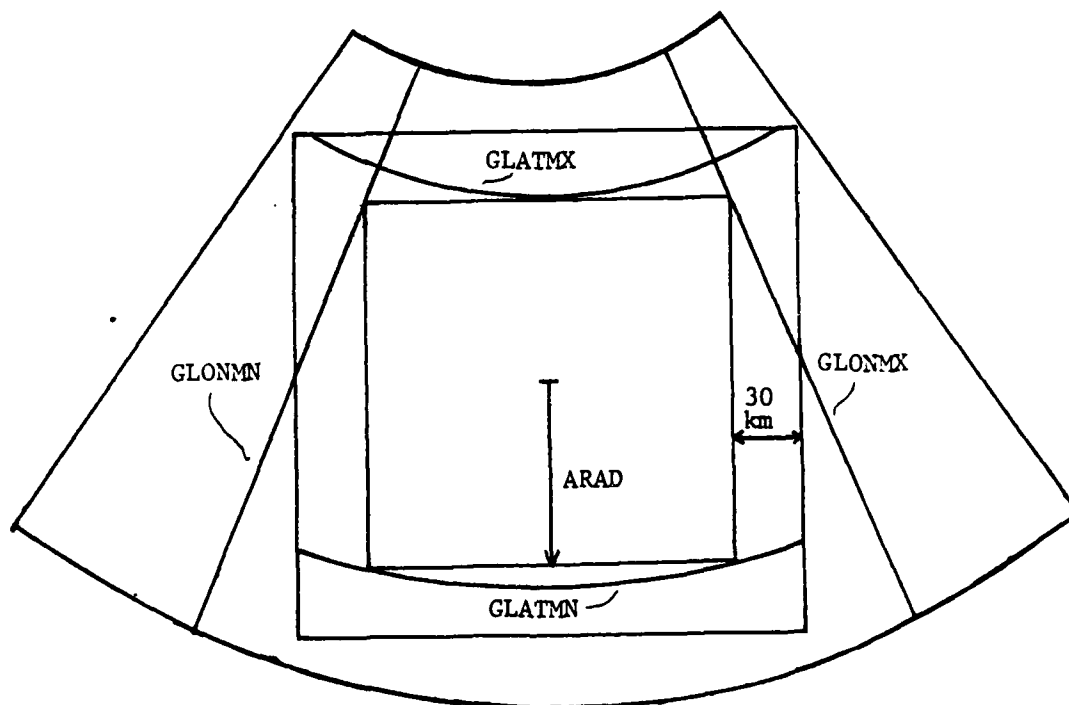
FIGURE A.1. Geographic region used for grid construction.

Data files used as input for GENBOT are digitized represenations of an NRL bathymetry chart [4] and were recorded on Mobil Oil Company Tape NAV-78 for use in a computer program to draw portions of the chart in the form of bathymetric contour lines and their depth labels as they exist on the chart. The records are in the card image format and consist of three types:

- NAME – depth value coordinates

- CON – contour line header label

- Chart coordinates of points along the contour line

In the original form the numerical data on the tape consisted of latitude/longitude pairs, providing no means of determining the depth represented by the contour lines without visual inspection of the computer plot. The tape used in this research is similar in format to NAV-78 except that the contour line header records include depth data (provided by NRL), and only the regions shown in Fig. A.2 and listed in Table A.I, are covered. The numerals of the data set names were chosen to reflect the original file number on the source tape, NAV-78.

Using the magnetic tape files selected by the user, GENBOT begins reading the digitized contour data, stopping after each record to discriminate the points, calculate geodetic distances, transform the data to x, y, z coordinates, and load the coordinates into three axis vectors, AX, AY, and AZ (which will be used later for interpolation) while simultaneously keeping a running plot of the contour lines.

After all input files have been processed, the vectors AX and AY are sorted into ascending order by the SSP3 subroutine RAIN1 [5]. The minimum and maximum values are tested to determine if the 30-km outside data margin is included in the range. If not, axis reduction begins until
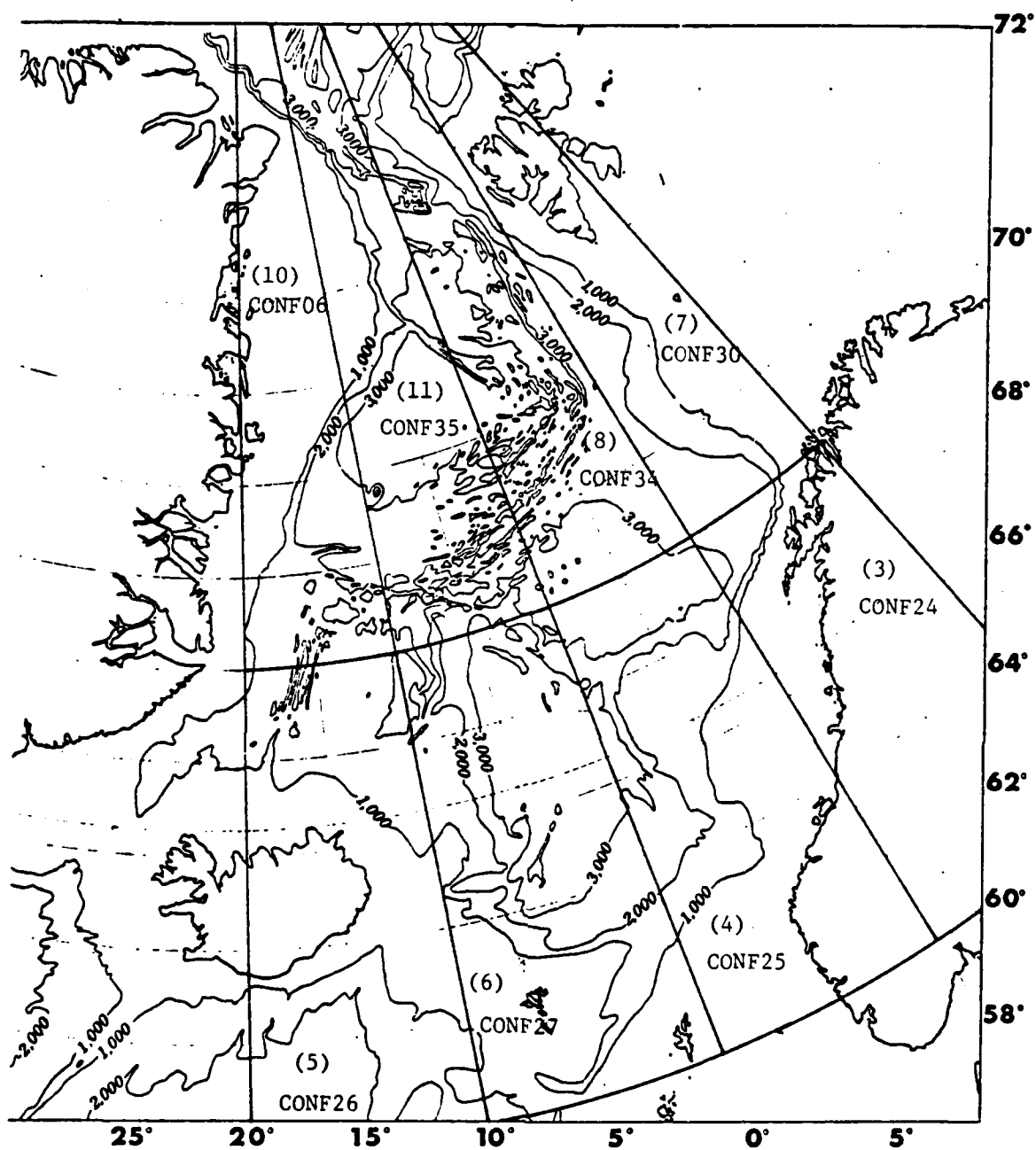
FIGURE A.2. North Atlantic regions, with their magnetic tape file numbers and data set names, available for use by GENBOT. All are 10° by 10°. Other regions available on the same tape are listed in Table A.I.

TABLE A.I

FILE ONE OF CONTOUR DATA RECORDED ON A MAGNETIC TAPE

DATA: BATHYMETRIC CONTOURS OF THE NORWEGIAN AND GREENLAND SEAS FROM
         LAT 60N TO 90N and LONG 20W to 20E.

SOURCE: ADAPTED FROM MOBILE OIL CO. TAPE NO. NAV-78 AND NAVAL RESEARCH
         LABORATORY DATA.

STRUCTURE: EACH FILE COVERS A 10 X 10 DEGREE REGION. THE FOLLOWING LIST

IDENTIFIES THESE FILES. THE FIRST VALUE IS THE FILE NUMBER, FOLLOWED BY

THE DSN, THEN THE LAT/LONG PAIR OF THE SOUTHWEST CORNER (LABOR,LONOR) OF

THE REGION, AND FINALLY THE NUMBER OF CARD IMAGES CONTAINED IN THE FILE.

THE TAPE IS STANDARD LABELED, DENSITY 1600 BPI, LRECL=80,BLKSIZE=800.

2	CONF13	+80-20	256
5	CONF26	+80-20	4343
8	CONF34	+70+00	12443
11	CONF35	+70-10	9428
3	CONF24	+60+10	569
6	CONF27	+60-10	6001
9	CONF37	+80+00	491
12	CONF36	+80+10	409
4	CONF25	+60+00	3026
7	CONF30	+70+10	857
10	CONF06	+70-20	4454
13	CONF38	+80-10	972

either the margin requirement is met or goes below a 7.5 km semi-axis length, the latter case preventing grid construction although a contour plot may still be produced. Subroutine CORNER is then called again to print the corner coordinate table. If no adjustments were necessary, the second table will be identical to the first.

The interpolation process begins at the southwest corner of the grid and proceeds in a columnar direction (i.e., completing the range of x_i values for each y_j value) via successive calls to the SSP3 subroutine RAIN2. The accuracy of the interpolated values is proportional to the density of data in the given area of interest. Every ten values interpolated are written as a record on the output file. The process continues until the total number of matrix elements has been reached, at which time the output file is closed and the program is terminated.

3. **User Instructions**

    a. JCL and Program Deck.

The program section of a sample deck is printed below. The TIME and REGION.GO parameters were chosen in anticipation of a maximum number of 5000 data points in the area of interest, a square region of 210 km on a side (150 km plus margin). The program listings included in this appendix are part of this example.

    //(Standard job card),TIME=20

    // EXEC FORTCLGW,REGION.GO=250K (See Appendix O.)

    // FORT.SYSIN DD *

        (GENBOT, CORNER, GEOPLT, GEODST listings)

        (RAIN1, RAIN3, RAIN2 listings)

    /*

The following DD statements identify the magnetic tape files representing the ten by ten degree regions into which the subroutine

extends. In the example, the area chosen is centered at coordinates 70°-04'N/15°-51' (70.07059/15.85302) with a radius of 75 km. This area requires two data files as input, CONF24 (file 3) and CONF30 (file 7). Since tape data is sequential, the order of the DD cards should be in the order of the file numbers (appearing in the LABEL parameter).

```
//GO.FT01F001 DD UNIT=34u0-3, VOL=SER=NPS,DISF=(OLD,PASS),
//    LABEL=(3,SL,,IN),DSN=CONF30
//GO.FT01F002 UNIT=AFF=FT01F001,VOL=(,RETAIN,REF=*.GO.FT01F001),
//    DISP=(OLD,PASS),LABEL=(7,SL,,IN),DSN=CONF30
```

More information on magnetic tape usage at NPS may be found in Ref. 10.

Next must follow the DD statements specifying the data set and device for permanent storage of the generated grid. The permanent device may be another magnetic tape, a disk, or punched cards. In all cases, the size of the output data set is important, and the DCB parameters should be considered in advance if the grid size is different from the example. GENBOT produces card image out put, meaning the logical record length is 80 bytes.

- Punched cards. If the grid is to be stored on punched cards, the user need only change the data set reference number on the appropriate WRITE statements in GENBOT from a two to a seven. The number of cards punched will equal a 3+[(row dimension)(column dimension)/10] rounded up to an integer value. This represents three header cards followed by ten depth values per card in columnar order, as required by ECTRACE. Thus if the input grid radius was 75 km and input cell resolution 1 km, the maximum parameter calculated by GENBOT will be:

XMAX	YMAX	IMAX	JMAX
150	150	151	151

resulting in a matrix of 22801 elements and a punched output of
2284 cards.

• <u>Disk</u>.   Grid storage on a disk device is the most convenient form for
multiple ECTRACE runs over the same sea bed.   The following DD state-
ment stores the example grid on a 3330 disk drive device at NPS as a
data set named S999.GRID.

```
//GO.FT02F001 DD UNIT=3330,VOL=SER=DISK03,DISP=(,KEEP),
//    DCB=(FECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(TRK,20),
//    DSN=S9999.GRID
```

Blocked at 800 bytes, the output data requires 14 tracks per 10
records (card images).   More information on the DCB and DSN parameters
may be found in Ref. 11.   When the generated grid is no longer needed,
and if the user wishes to generate a new grid, or if space has already
been allocated on the device, the DISP parameter should be SHR in
place of (KEEP) and the DCB and SPACE specifications may be omitted.
Disk data sets are periodically purged by center operations personnel,
but obsolete data set space should be cleared or reallocated by the
user.

    b.   Input Data.

       The following input data are user selected values and follow
the //GO.SYSIN DD* card.

Card	Variable	FORTRAN Field Descriptors	Remarks
1-2	TTL	6A8/6A8	Title
3	CNLAT, CNLON, DHN, ARAD	4F10.5	center latitude, center longitude, cell resolution, radius
4	NFILES, LEN	2I5	Number of input files, plot length

The magnitude of LEN determines the plot size (≤19 inches). If LEN is negative, no grid will be constructed. If its magnitude is less than four, no plot will be produced.

    c. Printed Output

GENBOT printed output supplies the following:

- A repeat of the user input data.

- A table of the grid corner coordinates and the depth matrix dimensions using the input parameters.

- A summary of results for each contour line, including its depth, and the title from the header label, the number of points on the line (NPTS) and the number of points used from the line (KPTS).

- A corner coordinate table of the final grid.

- A summary of the interpolation results. A line is printed for each interpolated point whose accuracy may be questioned. The meaning of the IER parameter may be found in the RAIN1 listing in Appendix I.

    d. Plotted Output

The contour lines of the input file are plotted with the local parallels and meridians overlaid. The resultant subregion represented by the output grid is outlined by a rectangle with labeled axes. The axis values are referenced to the desired grid origin and not the resultant grid origin, the latter being at (0,0) always.

    e. Programming Notes on Contour Data Interpolation

Subroutine RAIN2, the interpolation routine used by GENBOT, is aptly suited to interpolate points that are more or less randomly spread in both horizontal directions. The nature of contour data, however, is typically a collection of closely spaced depth values along widely spaced

contour (compare Figures III.1. and A.1.). This type of data arrangement frequently results in a colinearity problem for RAIN2, which can be resolved only by searching ever widening data cells for a non-colinear point, resulting in a degradation of accuracy. In addition, the routine is highly sensitive to contour line kinks which result in right-angle deviations in the interpolated contour lines (which may be plotted by GRDCHK).

Correcting the deficiencies of an interpolating routine is beyond the scope of this research, but the Naval Research Laboratory, Arctic Section, is currently investigating methods for generating grid-matrices from contour data. Users are encouraged to modify the code used in GENBOT as desired to produce results which more closely reflect the actual data supplied.

4. Input Contour Data Format

As stated previously, some contour data already exists in a format usable by GENBOT, but if data is supplied by another source, it must be arranged in a file in the manner and format specified below. All records are 80 byte card images.

The first record of the file contains data on the file itself. GENBOT uses a 4I5 field to represent the origin latitude/longitude pair, the number of contour lines in the file, and the number of card images it may expect to read.

Contour line data exists as separate sequences of cards and may appear in any part of the file as long as each line or line segment is in an unbroken sequence with a header card. The header card must contain the line's depth in hundreds of meters, right justified in a I4 field in columns 41-44. The cards following the header card contain the latitude/longitude pairs in degrees and decimals of degrees of the points along

71

the contour line in the form of four pairs of coordinates (in FORTRAN fields 8F10.5) per card. South and west values are negative. A value greater than 90.0 in magnitude in any latitude field is interpreted as the end of data for the contour line, and GENBOT expects the next record to be either another contour line header label, the end of the file, or a depth value position record.

Depth value position records are optional and are used during the chart-drawing phase of the program to place alphanumeric symbols on the chart product. They may exist any place after the first record in the file except imbedded in a contour line sequence. Columns 4-8 must contain "NAME" in an A4 field, columns 19-38 contain the latitude and longitude of the symbols in a 2F10.5 field and the symbols themselves must be in columns 46-47. These records are normally used to draw depth values in hundreds of meters on the contour lines.

If the user wishes to use GENBOT on externally supplied contour data, it it·recommended that a dump be performed on one of the smaller files (such as CONF13) of the GENBOT data tape to verify the required format for adaptation.

5. Output Grid Data Format

GENBOT constructs the output file in accordance with GRDSCT requirements. Since GRDSCT is the subroutine used by TRACER (of ECTRACE), G3DP and GRDCHK, any externally supplied grid data must all be of the same structure as outlined below in order to be used by those programs. Again, all records are card images.

a. The first two records contain the input grid title in the 10A8 fields.

b. The third record contains the input latitude and lontitude of the

grid center, the point spacing, and the number of rows and columns of the matrix representing the grid. The record format is (3F10.5,2I5).

    c. The fourth and remaining records contain ten depth values per card, in kilometers, as 10F8.4 fields, beginning with column one (y = 0) and proceeding all x-values before going to the next column, in accordance with the FORTRAN convention of columnar storage of matrices.

APPENDIX B
SYNGEN DESCRIPTION AND USER INSTRUCTIONS

1. Program Description

    a. SYNGEN is a collection of simple functions which may be selectively used to generate and record a matrix of depth values representing a desired synthetic bottom configuration. Six bottom cases are provided (see below), but the program may be easily modified by the user to produce a case not described here. A program listing is provided in Appendix H.

        The user selected parameters are:

        • CNLAT - any desired center latitude value.

        • CNLON - any desired center longitude value. These parameters are requested in compliance with the required output file format.

        • DMIN - the minimum depth value (D_1) to be attained within a 75 km radius from the grid center.

        • DMAX - the maximum depth value (D_0) within the same radius.

        • ITYPE - an integer code for the bottom type desired.

    b. Table B.1 lists the ITYPE code, the bottom type, the functions used to generate the depth matrix, and the matrix dimensions. All grids represent a region covering a 150 km in both the x and y directions, but the matrix size will be 11 by 11 or 151 by 151, depending upon the bottom type (DHN values are 15 km and 1 km, respectively). In all cases, $\Delta D = D_1 - D_0$.

    c. The depth file provided by SYNGEN conforms to the format required by ECTRACE, GRDCHK, and G3DP. When one of these programs uses a SYNGEN product, the array dimension in the program's source code must be checked to conform to the dimensions of the generated matrix.

74

TABLE B.I
SYNTHETIC SEA BED FUNCTIONS

ITYPE	Type	Function	Parameters	Matrix Dimensions
12	Wedge	$H = D_1 - \gamma$	$\gamma = \Delta D/135$ $0 \leq x \leq 135$	11 X 11
13	Ridge	$H = D_0 + \|\gamma\delta(x)\|$ $H = D_0$ elsewhere	$\gamma = \Delta D/15$ $\delta(x) = x - 75$ $60 \leq x \leq 90$	11 X 11
14	Trough	$H = D_1 - \|\gamma\delta(x)\|$ $H = D_1$ elsewhwere	$\delta = \Delta D/15$ $\delta(x) = x - 75$ $60 \leq x \leq 90$	11 X 11
15	Conical Seamount	$H = D_0 + \gamma r$ $H = D_0$ elsewhere	$\gamma = \Delta D/75$ $r = \sqrt{(x - 75)^2 + (y - 75)^2}$	151 X 151
16	Sinusoidal Undulation	$H = D_0 \gamma[1 + \cos\{\delta(x)/R\}]$	$\gamma = \Delta D/2$ $\delta(x) = 2(x - 75)$ $R = 15$	151 X 151
17	Parabolic basin	$H = D_1 - \gamma r^2$	$\gamma = \Delta D/(75)^2$ $r = \sqrt{(x - 75)^2 + (y - 75)^2}$	151 X 151

2. <u>User Instructions</u>

THE JCL of a sample deck are arranged as follows:

//(Standard JOB card),TIME=2

// EXEC FORTCLG, REGION.GO=250K

//FORT.SYSIN DD *

   (SYNGEN source code)

/*

//GO.FT02F001 DD UNIT=3330,VOL=SER=DISK03,

// DSN=S9999.SYNBED, DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),

// DSP=(NEW,KEEP), SPACE=(TRK,20)

//GO.SYSIN DD *

   (user selected data)

/*

Notice that the output data set is directed to DISK03. The DSN is the
user's selection for the data set name in accordance with NPS system re-
quirements. A 151 x 151 depth matrix requires twenty tracks of disk
space on a 3330 device.

Only one data card is needed to supply the program with the user's
parameter selections. Its field descriptor is (4F10.5,I5) and contains
the values reflected in the same order as listed in paragraph 1.a.

APPENDIX C
G3DP DESCRIPTION AND USER INSTRUCTIONS

1. Program Description

a. G3DP uses the NPS library subroutine PLT3D1 [Ref. 12] to draw a perspective surface plot of a subset of a depth matrix. This plot is a normal projection of the surface using a focal length and line-of-sight calculated from the user's selections for the viewpoint position angles DEGUP and DEGCW. As the variable names imply, the viewer is elevated DEGUP degrees from a reference depth and rotated DEGCW clockwise from a south-to-north viewing aspect. For example, the standard working matrix (dimensioned 151 by 151) is best viewed with DEGUP and DEGCW values of 30 degrees and 45 degrees respectively, providing the same viewing aspect demonstrated in all the the G3DP products printed in this report. Using a DEGUP value of zero results in an isometric rather than a perspective projection. The DEGCW selection should be between zero and 89 degrees (south-north to west-east viewing aspects).

b. As in ECTRACE and GRDCHK, G3DP requires source code modification by the user to tailor the program to the desired dimensions of the working matrix. Instructions in this regard are included as comments in the FORTRAN listing (Appendix K).

c. An important characteristic of G3DP is that it uses only the odd-numbered rows and columns of the working matrix to produce the surface plot. This reduction in resolution is required to reduce core requirements and unclutter the graphic output caused by typically large (greater than 75 by 75) working matrices.

2. User Instructions

2. <u>User Instructions</u>

The JCL of a sample program deck are arranged as follows:

```
//(Standard JOB card),TIME=2

//EXEC FORTCLGW,REGION.GO=300K   (Appendix O.)

//FORT.SYSIN DD*

     (G3DP source code)

/*

//LINK.USDD UNIT=3330,VOL=SER=DISK01,DSN=1570. ECTRACE,

//  DISP=SHR,LABEL=(, , ,IN)

//LINK.SYSIN DD*

    INCLUDE USDD(GRDSCT)

    ENTRY MAIN

/*

//GO.FT01F


//GO.SYSIN DD*

    (user-selected data)

/*
```

The LINK step provides access to GRDSCT for loading the working
matrix from the input file. The input data set is the same example used
in ECTRACE and GRDCHK. The card input data are user selected and con-
tained on one card, as follows (FORTRAN field descriptor 5F8.3)

- XST - working grid matrix x-origin relative to that of the
  input grid (km).

- YST - working grid-matrix y-origin relative to that of the input
  grid (km).

- DEGUP - viewpoint elevation angle ($0 \leq DEGUP \leq 89$ degrees).

- DEGCW – viewpoint rotation angle from south to north

  ($0 \leq$ DEGCW ≤ 89 degrees).

- LW – width of the plot ($5 \leq$ LW ≤ 18 inches).

APPENDIX D
GRDCHK-USER INSTRUCTIONS AND OUTPUT DESCRIPTION

1. Purpose and Output Description

Cases may arise where the accuracy of depth values in a generated matrix may be in question because of data sparsity, interpolation deficiencies or other reasons. Program GRDCHK assists in evaluating the generated grid by printing and plotting selected columns of the matrix for comparison with that of the source contours. The contour plot may also be used on a light table with an ECTRACE or ECCOM horizontal-ray trace plot (Figs. 5, 6, and 7).

The NPS library routine CONTUR is called for producing the contour plot. Subroutine GEOPLR is a modified version of GEOPLT to accommodate the rotated plot produced by CONTUR..

2. User Instructions

To aid in this discussion refer to a computer listing of GRDCHK code in Appendix J. A listing of the CONTUR subroutine may be found in Ref. 6.

The following example applies to a standard 151 by 151 working grid. Below is a listing of the JCL cards and the position of the program deck.

```
//(Standard job card), TIME=5
// EXEC FORTCGLW,REGION.GO=350K
//FORT.SYSIN DD *
  (GRDCHK,CORNER,GEOPLT,GEODST source decks)
/*
```

The following DD statements link to a member of ECTRACE for allocating a working subregion of the input.

```
//LINK.USDD DD UNIT=3330,VOL=SER=DISK01,
```

.

80

```
//  DSN= 1570.ECTRACE,DISP=SHR,LABEL=(,,,IN)

//LINK.SYSIN DD *

    INCLUDE USDD (GRDSCT)

    ENTRY MAIN

/*
```

Next follows the DD statement identifying the depth matrix input file. The same example is used for the GENBOT output file.

```
//GO.FTOFOO1 DD DSN=S9999.GRID,UNIT=3330,VOL=SER=DISK03,

//   DISP=SHR, LABEL=(,,,IN)
```

Finally the user input parameters are included. The card

```
//GO.SYSIN DD *
```

is followed by:

Card	Parameter Names	FORTRAN field description
1	NL, LEN	2I5
2	ZST, ZEND, XMIN, YMIN	4F6.1
3	IPS, IPE, JPS, JPE	4I5

where

- NL is the number of contour levels desired. The shallowest and deepest levels, if known, should be read into ZST and ZEND. If the boundary levels are not known, NL should be made negative, and subroutine CONTUR will calculate the boundary and contour levels. If NL is greater in magnitude than 100, no contour plot will be produced.

- LEN is the length of the contour plot's longest side in inches. If LEN is negative, a cartesian grid will be drawn on the contour plot instead of geographic coordinate lines. If LEN is less than 4, no contour plot will be produced. The maximum

81

magnitude of LEN is 19 inches.

- ZST AND ZEND are estimates of the smallest and largest values in the working matrix (ZB).

- XMIN and YMIN establish the origin of the working matrix relative to that of the input matrix. The program assignments for KMX and KMY, the recorded cell spacing DHN, and the dimensions of the working matrix determine the subregion size parameters in the same manner as they do in ECTRACE, since subroutine GRDSCT is linked to this program. In effect, the user must tailor certain portions of the program cards to compensate for the FORTRAN G inability to perform object time dimensioning.

- IPS, IPE, JPS and JPE identify the beginning and end rows and columns of the working depth matrix to be singled out for inspection. Thus, if ZB is dimensioned (150,150) and the above values are 1, 150, 74, and 78 respectively, GRDCHK will print the depth values of these 750 elements of the ZB matrix and plot five columnar profiles (Fig. 4). If this record is omitted, only the contour plot will be produced.

APPENDIX E
ECCOM OUTPUT DESCRIPTION AND USER INSTRUCTIONS

1. Description of Output

ECCOM allows the user to combine several selected ECTRACE runs and eliminate specific rays as desired to produce an improved horizontal ray path plot for investigation of curvature (Fig. 7 and Fig. 18). The size of the ECCOM plot will be the same size as first ECTRACE run of the input card deck unless the user opts to alter the scale factor (FACT) to control the plot dimensions.

The printed output is a list of all the rays processed for the ECCOM plot, giving each ray's initial elevation angle and heading. Additionally, the number of points used to plot each ray path is printed. This number of points corresponds to the number of reflections plus the starting point of the ray. Refer to Table VII for a sample listing of data for the ECCOM 51 plot, shown in Fig. 7.

2. User Instructions

The following JCL cards and parameters are required for an ECCOM run:

- JOB card – The CPU time parameter should allow four seconds for every 55 rays to be plotted.

- EXEC card – This card must specify FORTCLGW procedure.

- Calling program – Following the

    //FORT.SYSIN DD *

  is the ECCOM program deck. See Appendix L for the program listing.

- Data cards--Following the

    //GO.SYSIN DD *

  is the sequence of punched output decks, each individually arranged as described in Chapter V, Section C. The user may change the title

card (card # 1) of the first ECTRACE punched deck to identify the ECCOM run title. In addition, the user may change the scale factor (FACT) on card number 4 to modify the plot size. FACT was calculated by ECTRACE in response to the user's selection for the horizontal plot length (Chapter V, Section B). The following are sample values of FACT and recommended limits:

FACT	Horizontal Plot Axis Length (inches)
0.333	5 (minimum recommended)
1.000	15
1.267	19 (maximum recommended)

If the user desires a plot size outside the range recommended, refer to Ref. 7.

Cards 1 through 4 of each ECTRACE deck after the first are ignored by ECCOM, but the user should verify that they all reference the same portion of the same input grid-matrix in order for the ECCOM plot to be meaningful.

APPENDIX F
ANALYTICAL VERIFICATION OF ECTRACE HORIZONTAL RAY PATHS

The accumulation of ray heading changes after repeated specular reflection from a sloping sea bed describes a faceted horizontal path whose apparent curvature may in some instances be analytically described. Harrision [2] derived some solutions for horizontal ray paths over simple bottom topography cases. Most of these cases may be modeled by SYNGEN and used to test the validity of the horizontal ray paths mapped by ECTRACE. We chose an isovelocity sound speed condition and a conical seamount because its surface geometry effectively challenges the facet method of topographical modeling.

For a conical seamount of arbitrary slope but whose apex just touches the sea surface, Harrison gives the ray paths in polar coordinates (r, Φ) as

$$r = r_0 (1 - \cos^2\theta_0 \cos^2\phi_0)^{\frac{1}{2}}$$

$$\cdot \operatorname{cosec} \left| \frac{(1 - \cos^2\theta_0 \cos^2\phi)^{\frac{1}{2}}}{\sin\phi_0 \cos\theta_0} (\Phi + \Phi') \right| , \qquad (F.1)$$

where r_0 = initial distance from the apex

θ_0 = initial deviation angle

ϕ_0 = initial azimuthal angle

Φ' = a constant given by setting $\Phi = 0$ when $r = r_0$,

$$\Phi' = \frac{\sin\phi_0 \cos\theta_0}{(1 - \cos^2\theta_0 \cos^2\phi_0)^{\frac{1}{2}}} \sin^{-1}(1 - \cos^2\theta_0 \cos^2\phi_0)^{\frac{1}{2}} \qquad (F.2)$$

85

Figure F.1. is a diagram of a representative ray path. α is the deflection angle and represents the asymptotic angle of the shadow zone of the ray fan.
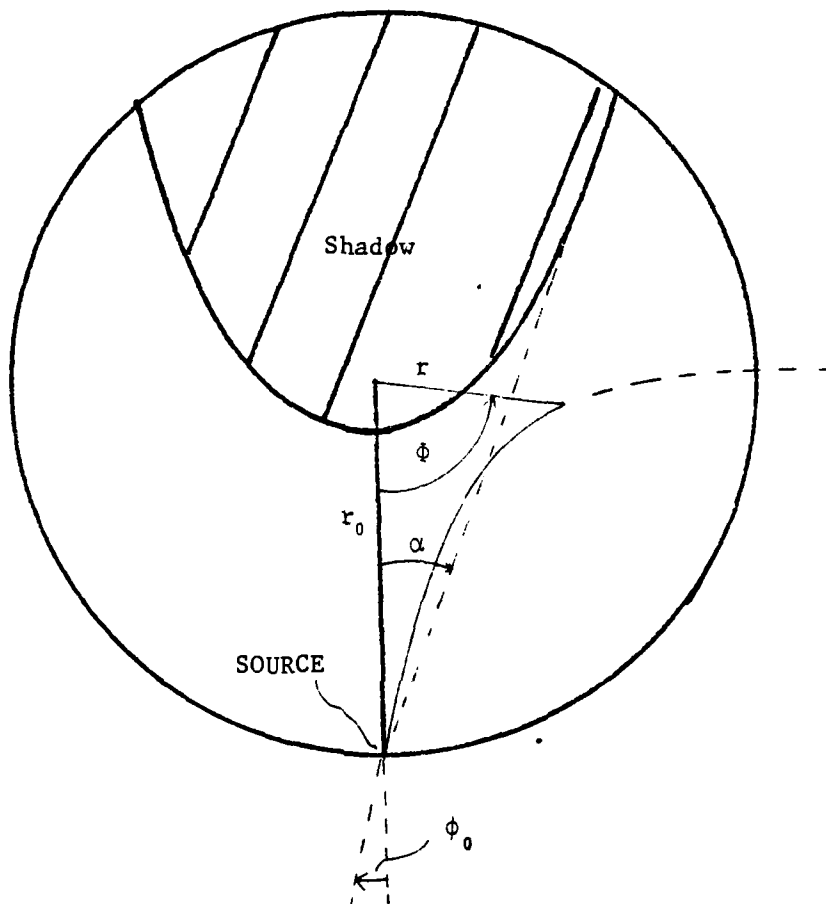
FIGURE F.1. Polar representation of the horizontal projection of a ray path for a conical seamount.

In the test run (Trial 83, Fig. F.2), the source was positioned at a range r_o of 50 km from the apex, bearing 180°, at a depth just below the surface. The receiver was placed at the apex so that for each ray the ECTRACE range r_E could be read directly from the printed ray history table values for DIST TO REC and Φ is the reciprocal of BNG TO REC. A zero gradient was used.
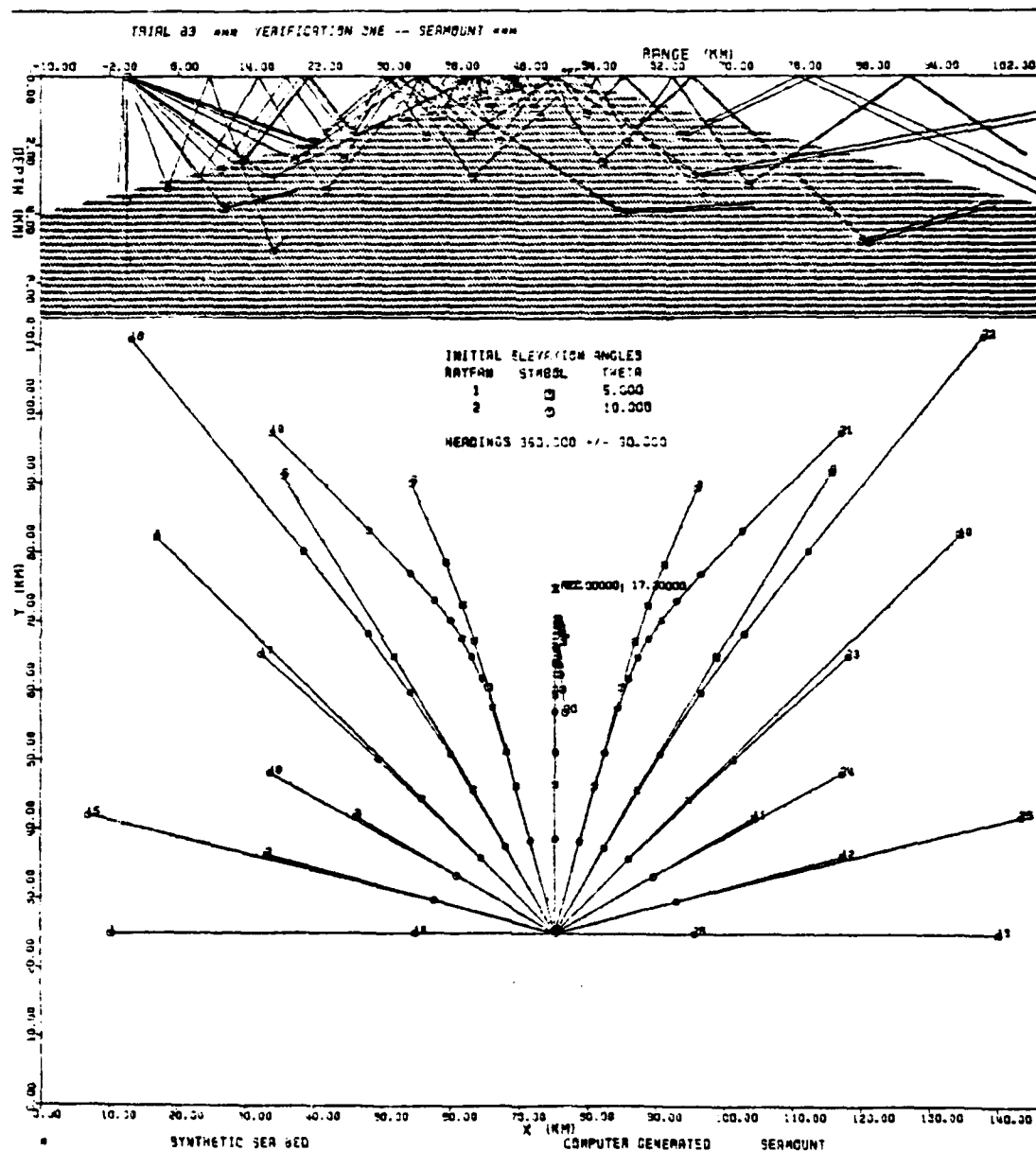
FIGURE F.2. Horizontal Plot of ECTRACE Trial for Harrison Verification.

Two ray lines were projected using $\theta_0 = 5$ degrees for ray fan 1 and $\theta_0 = 10$ degrees for ray fan 2. ϕ_0 ranged from 270° for the first ray to 90° for the last ray in each fan, with an increment of 15° between rays. Two rays were chosen for analysis, ray 6 ($\theta_0 = 5$ degrees, $\phi_0 = -15$ degrees) and ray 21 ($\theta_0 = 10$ degrees, $\phi_0 = 15$ degrees). The relevant portions of their trace histories are reproduced in Table F.I and Table F.II. Only the surface reflection values are used, where the slant range DIST becomes the horizontal range r_E. Inserting θ_0 and ϕ_0 values into Eq. (F.1), loaded in a programmable calculator, we obtained a table of values for r for both rays. The results are shown in Table F.III and Table F.IV alongside the ECTRACE values r_E and the observed error $\varepsilon = r - r_E$. These same results are plotted in Figures F.3 and F.4 (r vs Φ).

TABLE F.I

RAY 6 (6 OF RAY FAN 1) TRACE HISTORY

REV NO.	TYPES	DEPTH (KM)	GRID HEADING	ELEV ANGLE	BRNG TO REC	DIST FM REC
0	INIT	0.001	345.000	5.000	360.000	50.000
1	BOTT	1.943	344.500	-11.825	11.370	29.200
2	SURF	0.0	344.500	11.825	22.736	21.277
3	BOTT	1.153	343.010	-16.653	34.104	17.329
4	SURF	0.0	343.010	16.653	45.508	15.170
5	BOTT	0.934	340.647	-18.460	56.912	14.035
6	SURF	0.0	340.847	18.460	68.425	13.614
7	BOTT	0.919	338.231	-17.062	79.942	13.816
8	SURF	0.0	338.231	17.062	91.462	14.689
9	BOTT	1.096	336.557	-12.677	102.980	16.464
10	SURF	0.0	336.557	12.677	114.448	19.713
11	BOTT	1.729	335.915	-6.020	125.912	25.991
12	SURF	0.0	335.915	6.020	137.458	40.960

TABLE F.II

RAY 21 (8 OF RAY FAN 2) TRACE HISTORY

REV. NO.	TYPES	DEPTH (KM)	GRID HEADING	ELEV ANGLE	BRNG TO REC	DIST FM REC
0	INIT	0.001	15.000	10.000	360.000	50.000
1	BOTT	2.451	15.649	-17.119	354.386	36.838
2	SURF	0.0	15.649	17.119	348.770	29.483
3	BOTT	1.654	17.253	-23.468	343.153	24.866
4	SURF	0.0	17.253	23.468	337.518	21.761
5	BOTT	1.303	20.043	-28.702	331.883	19.590
6	SURF	0.0	20.043	28.702	326.243	18.046
7	BOTT	1.128	24.047	-32.594	320.603	16.949
8	SURF	0.0	24.047	32.594	315.014	16.200
9	BOTT	1.046	28.867	-33.973	309.425	15.723
10	SURF	0.0	28.867	33.973	303.767	15.480
11	BOTT	1.029	34.081	-33.685	298.105	15.459
12	SURF	0.0	34.081	33.685	293.479	15.661
13	BOTT	1.071	38.566	-31.042	286.852	16.097
14	SURF	0.0	38.566	31.042	281.203	16.802
15	BOTT	1.187	41.986	-26.620	275.55	17.834
16	SURF	0.0	41.986	26.620	269.889	19.295
17	BOTT	1.420	44.169	-20.799	264.224	21.345
18	SURF	0.0	44.169	29.799	258.538	24.279
19	BOTT	1.904	45.278	-14.066	252.851	28.620
20	SURF	0.0	45.278	14.066	247.160	35.467
21	BOTT	3.160	45.674	-6.733	241.469	47.511
22	SURF	0.0	45.674	6.733	235.781	73.529

90

TABLE F.III
Trial 83 Ray 6

ϕ	r	r_E	ε
0.0°	50.0 km	50.0 km	0.0km
-22.736	21.262	21.277	-0.015
-45.408	15.176	15.170	0.006
-68.425	13.616	13.614	0.002
-91.462	14.713	14.689	0.024
-114.49	19.822	19.713	0.109
-137.46	41.571	40.960	0.611

Ray 7 $r_{E_{ce}}$ = 4.339 km   ($\phi = \phi$, constant)

$r_{ce} - r_{E_{ce}}$ = 0.019 km

TABLE F.IV

Trial 83   Ray 21

Φ	r	r_E	ϵ
0.0°	50.0 km.	50.0 km	0.0 km
11.230	29.470	29.483	-0.013
22.482	21.745	21.761	-0.016
33.757	18.026	18.046	-0.020
44.986	16.178	16.200	-0.022
56.233	15.548	15.480	-0.022
67.521	15.643	15.661	-0.018
78.797	16.791	16.802	-0.011
90.111	19.309	19.295	0.014
101.46	24.365	24.279	0.086
112.84	35.822	35.467	0.355
124.219	75.857	73.529	2.238

Ray 20 $r_{E_{ce}}$ = 8.687 km   (Φ = 0, constant)

$r_{ce} - r_{E_{ce}}$ = -0.005 km
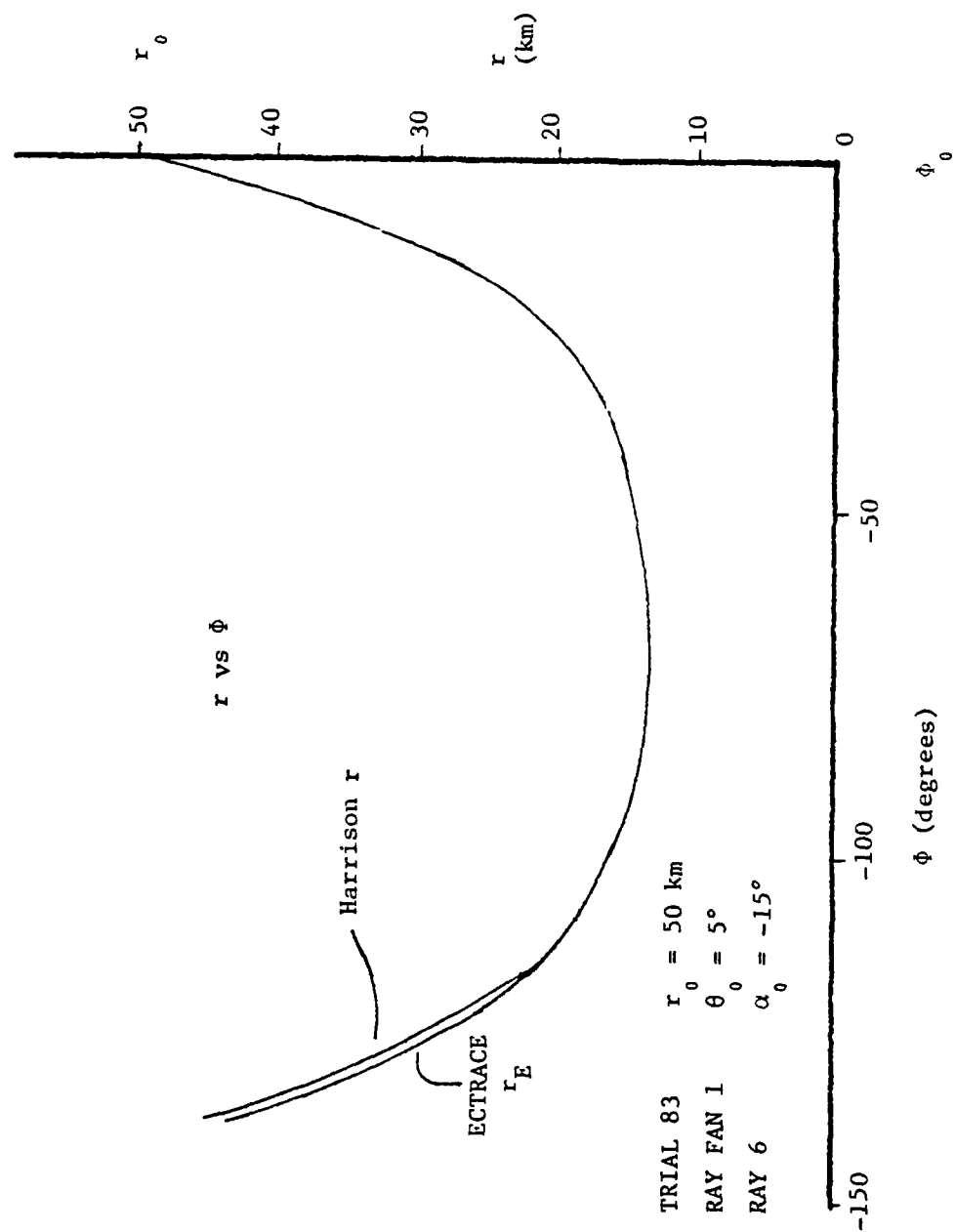
FIGURE F.3. Harrison verificaton plot of range vs Φ for ray 6.

93

FIGURE F.4. Harrison Verification Plot of range vs Φ for ray 21

These results show very close agreement up to a distance well past the closest point of approach, and only begin to diverge slightly when the rays approach their asymptotic values for Φ. Investigation of other rays of the same trial supports this observation. Clearly the fact that the plane facets do not all have the same slope as the cone they collectively represent, but vary as much as 4% between their extremes, explains most of the error. The validity of the model decreases near the apex, where fewer data points and hence plane facets are used to approximate the conical surface. In addition, the progressively shallower grazing angles assumed by the rays after CPA cause the horizontal paths to become more sensitive to this facet effect.

Reference 13 also provides an equation for the closest approach of the ray fan envelope to the seamount center, developed by letting $\phi_0 = 0$ in Eq. (F.1). Thus

$$
\begin{aligned}
r_{ce} &= r_0 \sin \theta_0 \\
&= 8.682 \text{ km.}
\end{aligned}
\tag{F.3}
$$

Ray 21 of Trial 83 was the radial ray ($\phi_0 = 0°$) and its CPA on the printout was 8.687 km, a difference of only four meters after 11 reflections from the bottom model. The ray should have reversed its heading exactly 180 degrees after CPA, but actually experienced a 174 degree heading change, revealing the sensitivity of the model to radially-directed ray paths.

Lastly, Ref. 13 defines the asymptotic angle of the shadow zone of the ray fan (Fig. F.1). A plot of the final azimuthal angle ϕ_f versus θ_0 for ray fan 2 (Fig. F.5) of the test run exhibits an interpolated value for α of 40 degrees for $\phi_0 = 29$ degrees. Testing this value with

OF **3**

A

B9931

α = Asymptotic deflection
angle for ray fan of
θ_0 = 10 degrees

ϕ_f vs ϕ_0

90

α

-45

45

α

-90
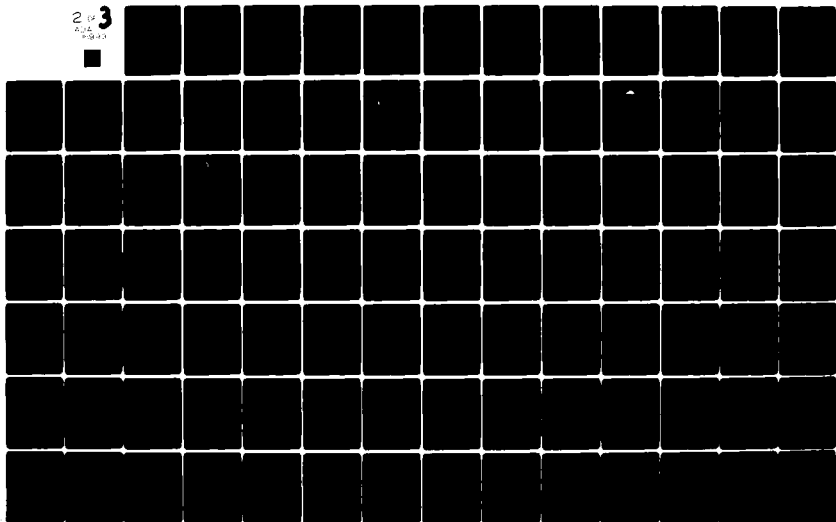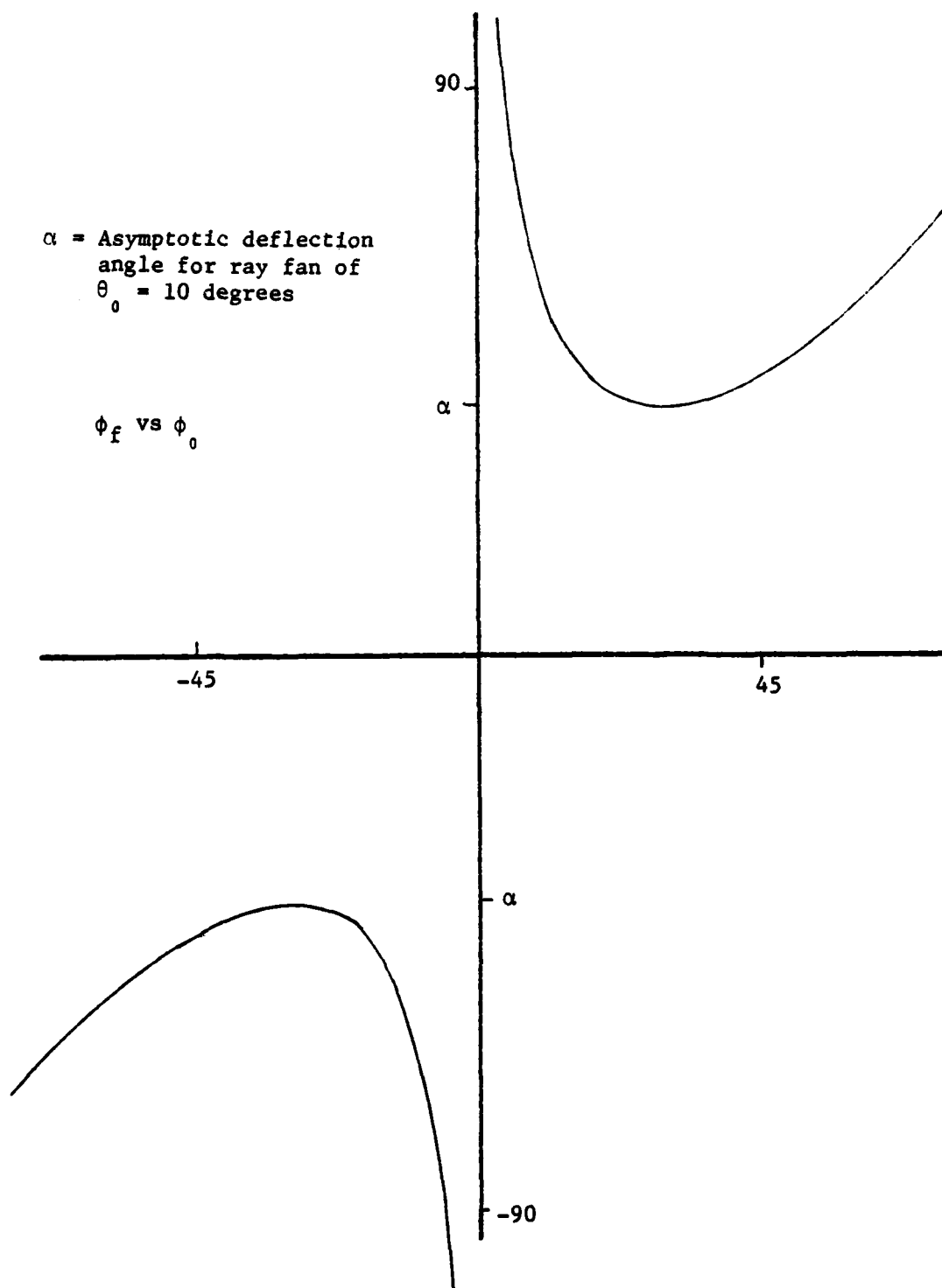
FIGURE F.5. Harrison verification plot of ϕ_f vs ϕ_0.

the formula

$$\alpha = \frac{3}{2}(\pi \sin^2 \theta_0)^{1/3}$$

$$= 39.187 \text{ degrees} \qquad\qquad (F.4)$$

shows close agreement with the theoretical value. Figure F.6, an ECCOM composite for the seamount, is a graphic example of the dependence of α on the initial elevation angle.

It is worth noting that the analytical expressions for horizontal projections of ray paths apply only to exact functional bottom topographies and that the curved three dimensional ray path surfaces only approximate an actual surface of vertical planes joined at bottom reflection points.
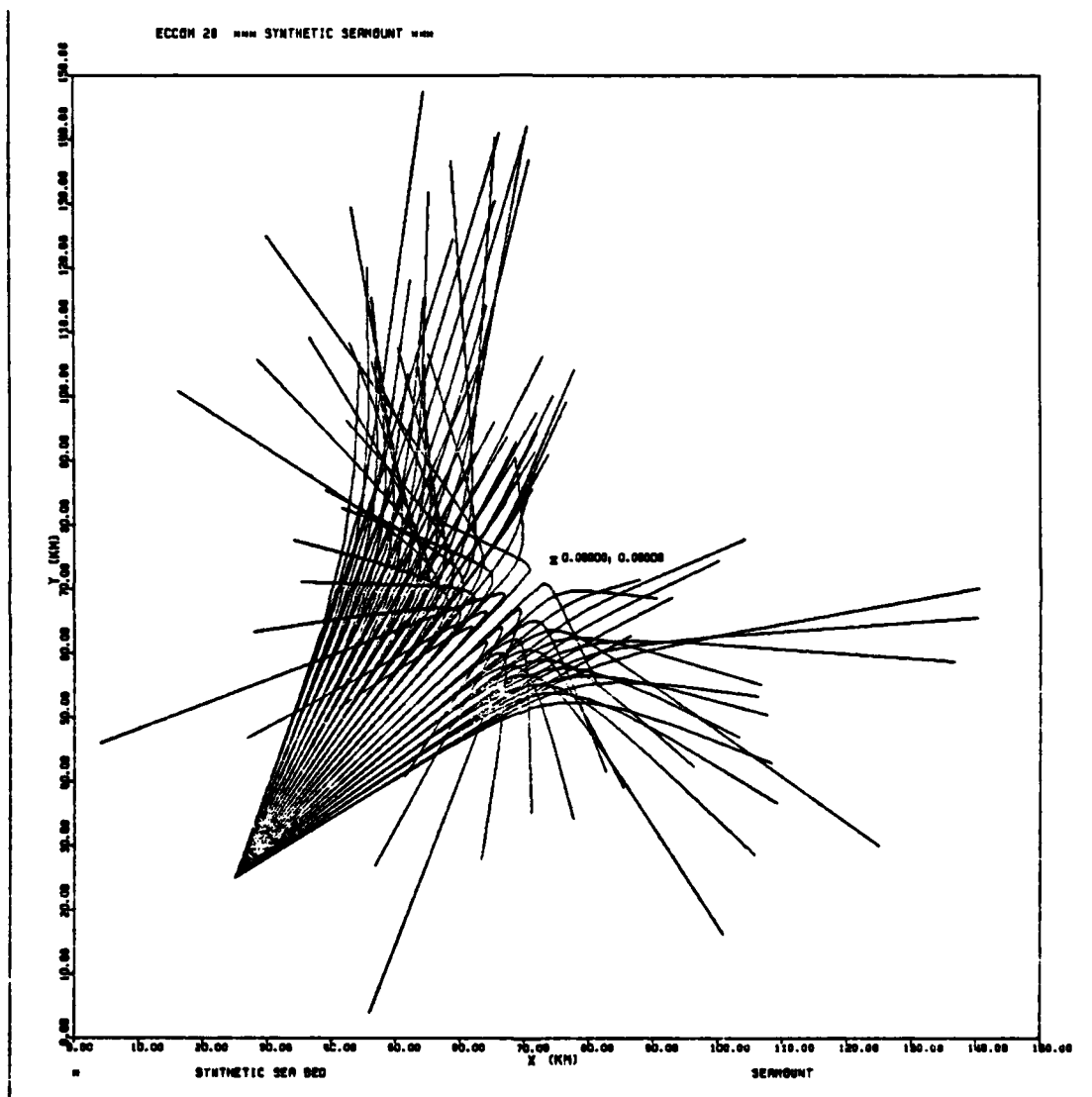
FIGURE F.6. ECCOM Trial 28 showing the effect of a synthetic conical
seamount on ray propagation. The ray's initial headings
varied from 20° to 58° and the initial elevation varied
from 4° to 18°. Selected rays from five ECTRACE runs
(100 rays) were combined.

98

APPENDIX G
TABLES
TABLE I
A SUMMARY OF THE PROGRAMS AND THEIR COMPONENT SUBPROGRAMS

I.   GENBOT - plots source contour data and generates depth matrix
     A.   CORNER - calculates boundary coordinates
     B.   GEOPLT - plots geometric grid lines
     C.   GEODST - solves geodetic triangle
     D.   RAIN 1/2/3 - perform sorting and interpolation (from SSP3)

II. SYNGEN - produces synthetic sea bed depth matrices

III. G3DP - produces perspective surface plot of depth matrix
     A.   GRDSCT - loads working surface plot of depth matrix
     B.   PLT3D1 - draws surface plot (from NPS)

IV. GRDCHK - allows inspection of depth matrix and draws contour plot
                of same
     A.   GRDSCT - see above
     B.   CORNER - see above
     C.   GEOPLR - same as GEOPLT but for a rotated plot
     D.   GEODST - see above
     E.   CONTUR - draws a contour plot of depth matrix (from NPS)

V.   ECTRACE - a collection of subprograms which perform ray tracing
     A.   TRACER - controls all subprograms and manages ray traces
     B.   GRDSCT - see above
     C.   IDTSUB - identifies subscripts of depth matrix
     D.   DEEP - calculates depth at given coordinates
     E.   IDPROF - identifies water mass
     F.   NUPROF - calculates water mass parameters
     G.   CHNLIM - calculates sound channel limits
     H.   CONTAC - calculates point of bottom contact
     I.   BOANG - calculates direction and grazing angle of bottom
                reflected ray
     J.   FBLOSS - calculates bottom loss
     K.   ANGPRT - prints ray fan bottom bounce table
     L.   BNGPLT - initializes plotting and draws plot borders
     M.   BDTPLT - draws bathymetry profile
     N.   RNGPLT - draws vertical plot
     O.   T2DPLT - draws horizontal plot
     P.   SSPPLT - draws sound speed profiles
     Q.   ENDPLT - draws plot legend and terminates plotting

     (L through Q call subroutines from the Versatec software package.)

TABLE II
INPUT PARAMETERS TABLEAU (COMPUTER PRINT OUTPUT)

ECTRACE TRIAL 67 *** SYNTHETIC BASIN ***

SOUND SPEED PROFILES

ZC	C(ZC)	GRAD
0.0	1461.08	0.0
50.00	1461.00	-1.40000
60.00	1447.00	-0.20000
70.00	1445.00	0.10000
90.00	1447.00	0.01780
500.00	1454.30	0.01218
1050.00	1461.00	0.02000
1500.00	1470.00	0.01668
4000.00	1511.70	0.01668

NWMR (Northwest Mohns Ridge)

ZC	C(ZC)	GRAD
0.0	1490.00	-0.16000
50.00	1482.00	-0.05000
200.00	1474.50	-0.00286
375.00	1474.00	-0.02133
750.00	1466.00	-0.00444
975.00	1465.00	0.00727
1250.00	1467.00	0.01200
1500.00	1470.00	0.01668
4000.00	1511.70	0.01668

SEMR (Southeast Mohns Ridge)

ZC	C(ZC)	GRAD
0.0	1470.00	0.0
1500.00	1470.00	0.01668
4000.00	1511.70	0.01668

ISOV (Isovelocity)

(Table II continued)

100

FRONT/BOUNDARY INTERSECTIONS

	FRONT			FRONT	
ENTRY		EXIT	ENTRY		EXIT
(0.0, 50.0)		(100.0,150.0)	(50.0, 0.0)		(150.0, 100.0)

GRID FILE CALLED
```
*******************************************************
*                                                     *
*               SYNTHETIC SEA BED                     *
*              COMPUTER GENERATED  BASIN              *
*                                                     *
*******************************************************
```
HAD BEEN READ. CENTER LAT/LONG IS 70.07059/ 15.85032 AT
(75.000, 75.000) KM. RANGES ARE 0.0 TO 150.0 KM IN Y (22801 ITEMS).

THE FOLLOWING ADDITIONAL PARAMETERS WERE INPUT:

SOURCE DATA: FREQUENCY 150.00 HZ,GRID COORDINATES (40.00,40.00),
DEPTH 0.850 KM. 8 RAY FANS WILL BE TRACED, FROM -30.0 TO 30.0 DEGREES
ELEVATION, INCREMENTING EVERY 8.6 DEG. 1 RAYS MAKE A RAY FAN FROM 10.0
TO 10.0 DEGREES TRUE HEADING, INCREMENTING EVERY 0.0 DEGREES.

RECEIVER DATA: GRID COORDINATES (125.0, 40.0), DEPTH 1.749 KM.

GRID ARRAY: MAXIMUM ARRAY SIZE (151,151).DEPTH VALUES RESOLVED BY 1.0
            KM. SPACING.

TRACE PARAMETERS: PROPAGATION WILL CONTINUE UNTIL A BOTTOM BOUNCE LOSS
                  OF 150.0 DB OR 21 BOTTOM CONTACTS EXCEEDED FOR EACH
                  RAY. PATH INCREMENT IS 100.0 M. POSITION ERROR AFTER
                  BOTTOM CONTACT WITHIN 0.05000 KM. CHANNEL RANGE
                  LIMIT 200.0 KM. SMALL ANGLE 2.0 DEG.

PLOT OPTION SELECTED.

PLOT PARAMETERS:	X	Y	Z	R
AXES:	0.0	0.0	-1.0	-10.0
	10.0	10.0	2.0	7.00
	150.0	150.0	6.0	123.00

        SCALING: LENGTH 4 IN., FACTOR=0.2667

EACH SEGMENT REPRESENTS 10 POSITION CALCULATIONS.

(program parameters are user supplied via DATA cards)

TABLE III

Ray History Tableau ECTRACE Trail 67 (computer print output)
Ray 8 (of Ray Fan 8) TRACE HISTORY

REV. NO.	TYPE	X(KM)	Y(KM)	DEPTH(KM)	HEADING	ELEV. ANG.
0	INIT	40.000	40.000	0.850	10.000	30.000
1	BOTT	40.722	44.097	3.161	12.533	-20.323
2	SURF	42.397	51.631	0.0	12.533	21.104
3	BOTT	44.611	61.591	4.043	14.157	-14.658
4	SURF	47.628	73.549	0.0	14.157	17.615
5	BOTT	50.933	86.652	4.344	15.556	-14.938
6	SURF	54.384	99.050	0.0	15.556	18.425
7	BOTT	57.154	109.001	3.628	17.472	-22.313
8	SURF	59.508	116.479	0.0	17.472	26.133
9	BOTT	61.260	122.046	2.780	20.496	-31.868
10	SURF	62.747	126.024	0.0	20.496	33.698
11	BOTT	63.904	129.119	2.190	25.415	-41.798
12	SURF	64.930	131.279	0.0	25.415	42.680
13	BOTT	65.778	133.064	1.823	33.460	-51.533
14	SURF	66.569	134.260	0.0	33.460	51.978
15	BOTT	67.260	135.305	1.606	46.953	-60.129
16	SURF	67.933	135.934	0.0	46.953	60.372
17	BOTT	68.551	136.511	1.490	69.313	-65.773
18	SURF	69.182	136.749	0.0	69.313	65.930
19	BOTT	69.787	136.977	1.452	97.778	-66.103
20	SURF	70.428	136.890	0.0	97.778	66.244
21	BOTT	71.073	136.802	1.482	121.284	-61.160
22	SURF	71.771	136.378	0.0	121.284	61.347
23	BOTT	72.510	135.929	1.587	135.723	-53.031
24	SURF	73.338	135.080	0.0	135.723	53.338
25	BOTT	74.265	134.129	1.786	144.372	-43.592
26	SURF	75.339	132.631	0.0	144.372	44.152
27	BOTT	76.618	130.845	2.127	149.526	-34.065
28	SURF	78.158	128.228	0.0	149.526	35.164
29	BOTT	80.046	125.020	2.664	153.024	-26.290
30	SURF	82.376	120.443	0.0	153.024	26.244
31	BOTT	85.445	114.414	3.452	155.238	-19.001
32	SURF	89.145	106.392	0.0	155.238	20.346
33	BOTT	93.855	96.179	4.247	156.763	-15.037
34	SURF	98.821	84.615	0.0	156.763	18.321
35	BOTT	103.807	73.002	4.261	158.297	-17.529
36	SURF	107.968	62.549	0.0	158.297	20.418
37	BOTT	111.257	54.284	3.486	160.557	-25.605
38	SURF	113.497	47.940	0.0	160.557	28.126
39	BOTT	115.211	43.084	2.713	164.103	-35.243
40	SURF	116.231	39.501	0.0	164.103	36.331
41	BOTT	117.049	36.630	2.188	169.812	-45.355

BOTTOM LOSS GATE EXCEED. TERMINATING RAYTRACE AFTER 3311
SEGMENTS. PLOT BASED ON 367 POINTS, 41 REVERSALS AND 21
BOTTOM BOUNCES. MINIMUM DISTANCE TO RECEIVER, 8.57 KM,
OCCURS AT TIME 166.244

(TABLE III continued)

102

continued TABLE III

BRING TO SRC*	DIST FM SRC*	TIME	WTR MASS	LYR	DEPTH MIN	LIMITS MAX
0.0	0.0	0.0	SEMR	5	0.0	*****
190.00	4.759	3.270	SEMR	8	0.0	*****
191.646	11.906	8.802	SEMR	1	0.0	*****
192.056	23.308	16.265	SEMR	9	0.0	7.093
192.809	34.416	24.931	SEMR	1	0.0	7.093
193.189	48.043	34.520	SEMR	9	0.0	7.526
193.690	60.783	43.604	SEMR	1	0.0	7.526
193.961	71.155	51.036	NWMR	8	0.0	*****
194.310	78.932	56.802	NWMR	1	0.0	*****
194.572	84.778	61.279	NWMR	8	0.0	*****
194.812	88.984	64.648	NWMR	1	0.0	*****
195.015	92.279	67.440	NWMR	8	0.0	*****
195.276	94.626	69.600	NWMR	1	0.0	*****
195.483	96.573	71.568	NWMR	8	0.0	*****
195.741	97.936	73.103	NWMR	1	0.0	*****
195.962	99.130	74.624	NWMR	8	0.0	*****
196.234	99.921	75.814	NWMR	1	0.0	*****
196.480	100.647	77.058	NWMR	7	0.0	*****
196.784	101.058	78.109	NWMR	1	0.0	*****
197.074	101.451	79.258	NWMR	7	0.0	*****
197.435	101.559	80.307	NWMR	1	0.0	*****
197.796	101.669	81.494	NWMR	7	0.0	*****
198.245	101.483	82.550	NWMR	1	0.0	*****
198.721	101.290	83.914	NWMR	8	0.0	*****
199.322	100.758	85.183	NWMR	1	0.0	*****
200.003	100.176	86.827	NWMR	8	0.0	*****
200.882	99.146	88.467	NWMR	1	0.0	*****
201.954	97.956	90.673	NWMR	8	0.0	*****
203.388	96.130	93.144	NWMR	1	0.0	*****
205.222	93.996	96.346	SEMR	8	0.0	*****
207.779	90.926	100.139	SEMR	1	0.0	*****
211.412	87.232	105.373	SEMR	8	0.0	*****
216.510	82.606	111.713	SEMR	1	0.0	*****
223.790	77.898	110.854	SEMR	9	0.0	7.469
232.820	73.832	128.722	SEMR	1	0.0	7.469
242.651	71.917	137.727	SEMR	9	0.0	8.687
251.646	71.615	145.718	SEMR	1	0.0	8.687
258.664	72.722	152.236	ISOV	2	0.0	*****
263.834	73.929	157.358	ISOV	1	0.0	*****
267.652	75.297	161.318	ISOV	2	0.0	*****
270.375	76.237	164.413	ISOV	1	0.0	*****
272.504	77.134	166.955	ISOV	2	0.0	*****

*For this run range and bearing of the rays were referenced to the source. As explained in Chapter V, in the current version of ECTRACE, these values are now referenced from the receiver (REC).

(TABLE III continued)

continued TABLE III

C (KM/S)	GRAZ ANGLE	BTM ANGLE	BTM LOSS	LPS
1.46556				
1.49770	24.081	4.678	4.1	0
1.49000				
1.51242	16.731	3.392	6.1	0
1.49000				
1.51743	14.465	2.756	7.2	0
1.49000				
1.50549	19.479	3.935	10.0	1
1.46100				
1.49135	27.785	5.015	15.1	1
1.46100				
1.48151	37.224	5.651	22.2	1
1.46100				
1.47539	46.949	6.015	30.0	0
1.46100				
1.47177	56.159	6.182	40.9	0
1.46100				
1.46980	63.522	6.272	50.9	0
1.46100				
1.46904	66.735	6.262	60.0	0
1.46100				
1.46963	64.227	6.248	70.9	0
1.46100				
1.47145	57.381	6.141	80.9	0
1.46100				
1.47477	48.418	6.040	89.9	0
1.46100				
1.48045	38.786	5.639	97.3	0
1.46100				
1.48941	30.571	5.162	103.0	2
1.49000				
1.50255	22.176	4.155	106.6	0
1.49000				
1.51582	16.287	2.896	108.3	0
1.49000				
1.51606	16.298	2.906	110.1	1
1.49000				
1.50313	22.375	4.256	113.8	0
1.47000				
1.49023	30.985	5.212	119.5	0
1.47000				
1.48147	40.627	5.821	127.3	0

TABLE IV

Ray fan Summary Tableau for ECTRACE Trail 67 ray number 8. (computer print output)

ANGLES BEFORE BOUNCE.   INITIAL ELEVATION ANGLE = 30.00

HEADINGS:	10.00
BOUNCE	
1	27.78
2	18.77
3	13.96
4	16.60
5	23.63
6	32.51
7	42.10
8	51.66
9	60.18
10	65.80
11	66.12
12	61.15
13	52.96
14	43.36
15	34.76
16	25.30
17	17.50
18	15.03
19	19.09
20	26.64
21	35.75

DEPTH IN METERS AT BOUNCE.   INITIAL ELEVATION ANGLE = 30.00

HEADINGS:	10.00
BOUNCE	
1	3160.6
2	4042.9
3	4343.8
4	3627.9
5	2780.2
6	2190.1
7	1823.4
8	1606.3
9	1490.1
10	1452.0
11	1481.7
12	1586.9
13	1785.7
14	2126.5
15	2663.6
16	3451.5
17	4246.8
18	4261.3
19	3486.1
20	2713.1
21	2187.8

TABLE V

CPA Summary Tableau for ECTRACE Trial 67 (computer printed output)

8 RAY TRACES COMPLETED

H HEADING AT CLOSEST APPROACH TO RECEIVER
T TIME AT CLOSEST APPROACH TO RECEIVER
D MINIMUM DISTANCE TO RECEIVER

RAY NUMBER	INITIAL HEADING	INITIAL ELEVATION	NUMBER OF BOUNCES	H	T	D
1	10.000	-30.000	21	164.266	166.534	7.888
2	10.000	-21.429	21	158.372	165.018	1.339
3	10.000	-12.857	21	146.487	163.575	6.884
4	10.000	-4.286	21	145.103	160.945	7.175
5	10.000	4.286	12	144.811	160.593	7.803
6	10.000	12.857	21	149.200	163.395	6.676
7	10.000	21.429	21	157.810	163.22	1.139
8	10.000	30.000	21	164.103	166.244	8.575

TABLE VI

RAY FAN SYMBOLS – used to denote location of bottom reflections in the ECTRACE plots. Thirteen different symbols are used before repetition. All rays of the same ray fan (initial elevation angle) will indicate a bottom reflection with the same symbol. Each ECTRACE plot lists the symbols used for the horizontal and vertical plots.

INITIAL ELEVATION ANGLES

RAYFAN	SYMBOL	THETA
1	⊡	-10.000
2	⊙	-9.000
3	△	-8.000
4	+	-7.000
5	×	-6.000
6	◇	-5.000
7	⊕	-4.000
8	⊠	-3.000
9	Z	-2.000
10	Y	-1.000
11	⋈	0.000
12	✳	1.000
13	⊠	2.000
14	⊡	3.000
15	⊙	4.000
16	△	5.000
17	+	6.000
18	×	7.000
19	◇	8.000
20	⊕	9.000
21	⊠	10.000

HEADINGS 90.000 +/- 0.000

TABLE VII

ECCOM printed output. The rays which are plotted are listed, and their initial heading, initial elevation, and the number of points (initial plus reflections) are indicated. In this run the plot produced is shown in Fig. 7.

ECCOM 51 *** SYNTHETIC BASIN ***
SYNTHETIC SEA BED
COMPUTER GENERATED

XST	XDV	YST	YDV	CNX	CNY	BASIN CNLAT	CNLON
0.0	10.000	0.0	10.000	75.000	75.000	0.0	0.0

INTIAL ELEVATION	INTIAL HEADING	POINTS	SCALE FACTOR= 0.340
40.000	40.000	21	
40.000	44.000	21	
40.000	48.000	21	
38.000	42.000	21	
38.000	46.000	21	
36.000	40.000	21	
36.000	44.000	21	
36.000	48.000	21	
34.000	42.000	21	
34.000	46.000	21	
60.000	40.000	21	
60.000	44.000	21	
60.000	48.000	21	
51.333	42.000	21	
51.333	46.000	21	
42.667	40.000	21	
42.667	44.000	21	
42.667	48.000	21	
34.000	42.000	21	
34.000	46.000	21	
60.000	5.000	21	
60.000	9.000	21	
60.000	13.000	21	
62.000	7.000	21	
62.000	11.000	21	
64.000	5.000	21	
64.000	9.000	21	
64.000	13.000	21	
66.000	7.000	21	
66.000	11.000	21	

APPENDIX H
FIGURES

FIGURE 1. Program and Data Utilization Flowchart.
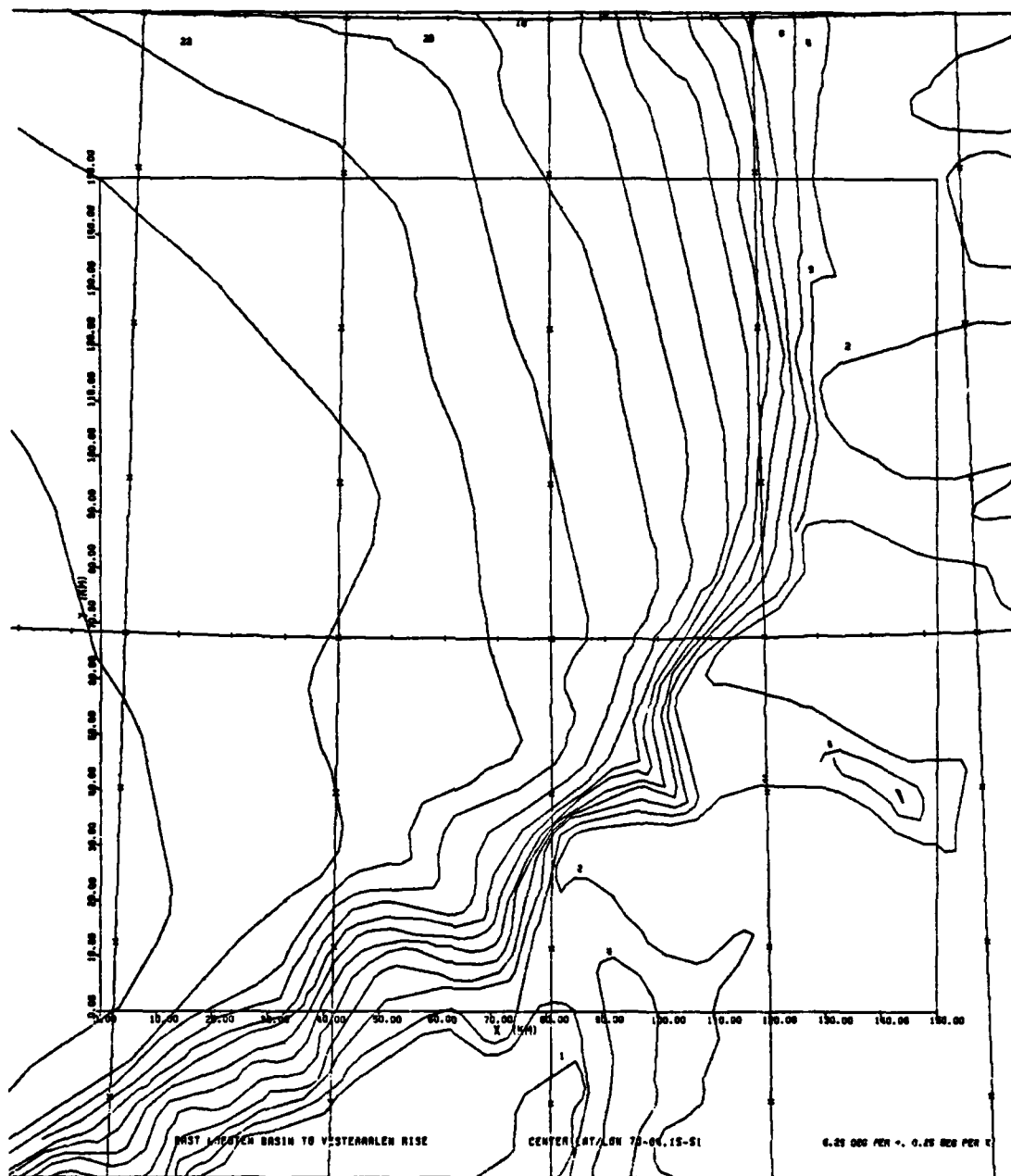
FIGURE 2. A contour plot of the eastern Lofoten Basin produced by
GENBOT from magnetic tape data. The rectantular labeled
axis denotes the region for which GENBOT generated a
matrix representing evenly spaced depth values. A
geographic coordinate grid is superimposed.

FIGURE 3.   ECTRACE TRIAL 82C performed over a depth-grid matrix of
            the Mohns Ridge. In this run three ray fans are used
            with eight rays per ray fan. This may be overlaid on
            a contour plot of this area (Fig. 2).
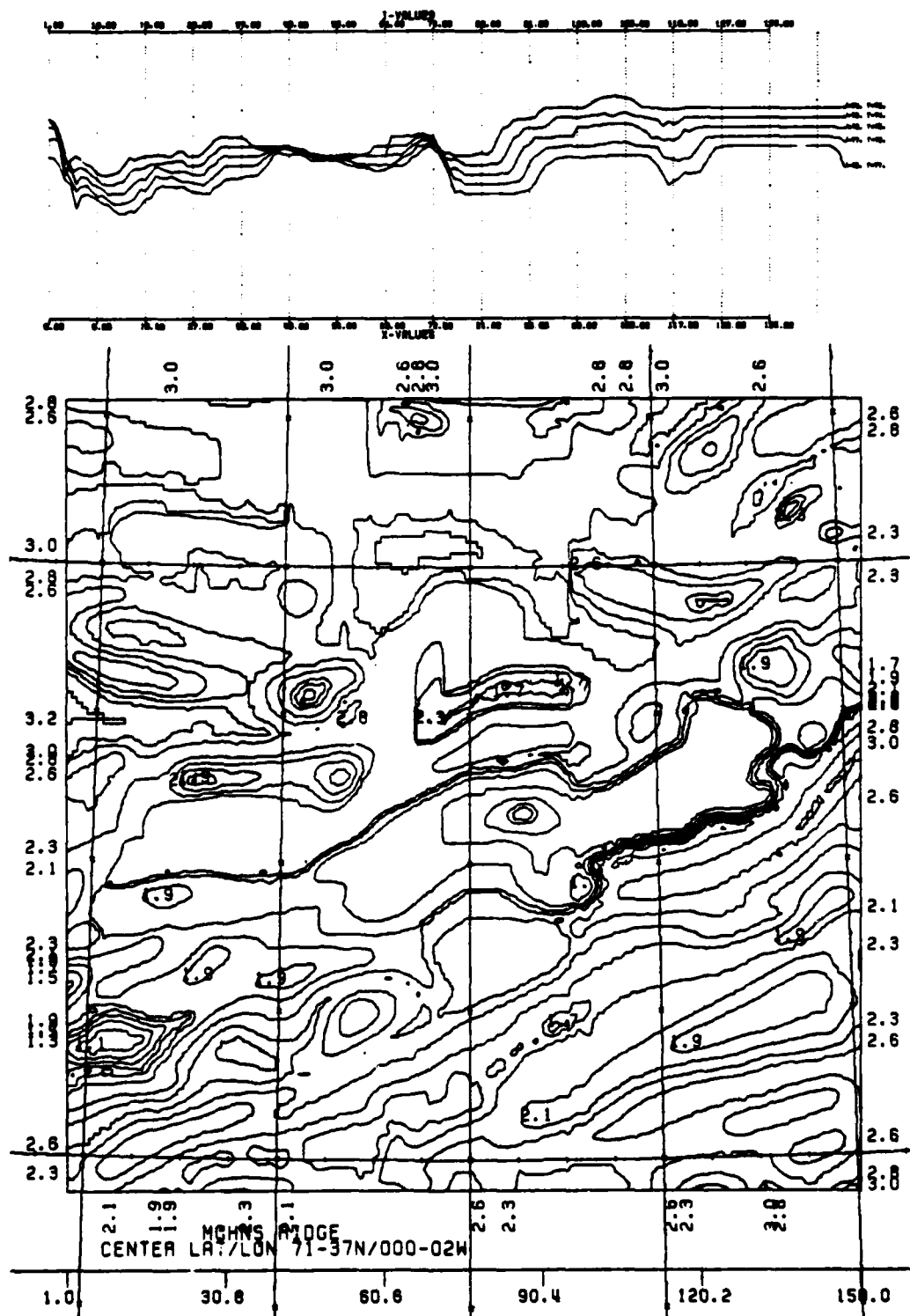
FIGURE 4.   A GRDCHK plot made on a matrix generated from contour data.
In this example, rows 1 through 150 of columns 74 through 78
were selected for profile analysis and printing.
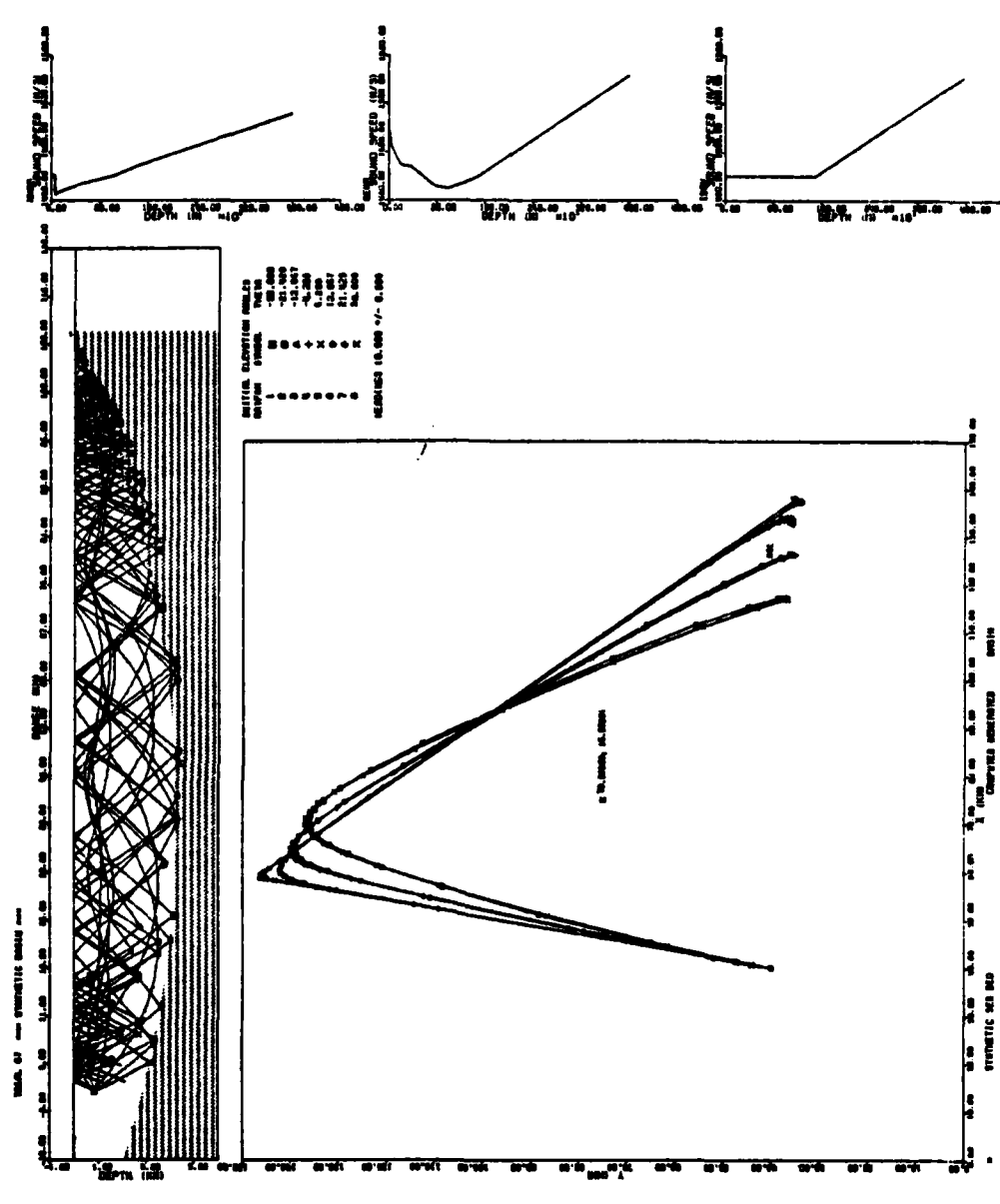
FIGURE 5.   ECTRACE TRIAL 67, performed in a synthetic parabolic basin
(FIGURE 6).   In this run eight ray fans are used with initial
elevation angles varying between -30 and +30 degrees. Since
there is only one ray in each ray fan, the initial heading
for the rays will be 10 degrees.   The vertical plot reveals
CZ, sofar ducted, and SRBR rays.

114
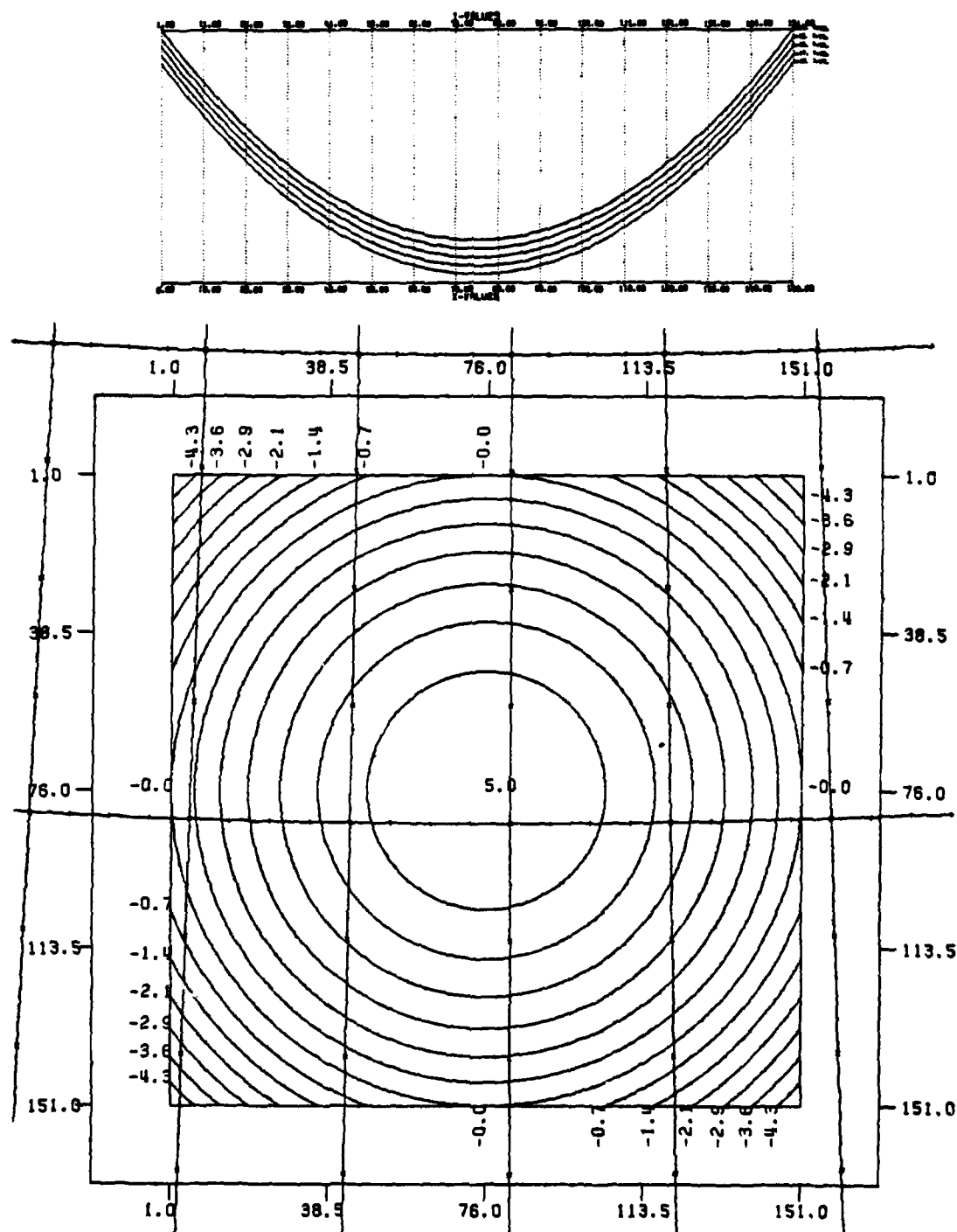
FIGURE 6. A graphical analysis performed by GRDCHK of a synthetic
parabolic basin (Fig. 10) created by SYNGEN. The upper
plot is a series of displaced depth curves of selected
columns of the matrix.
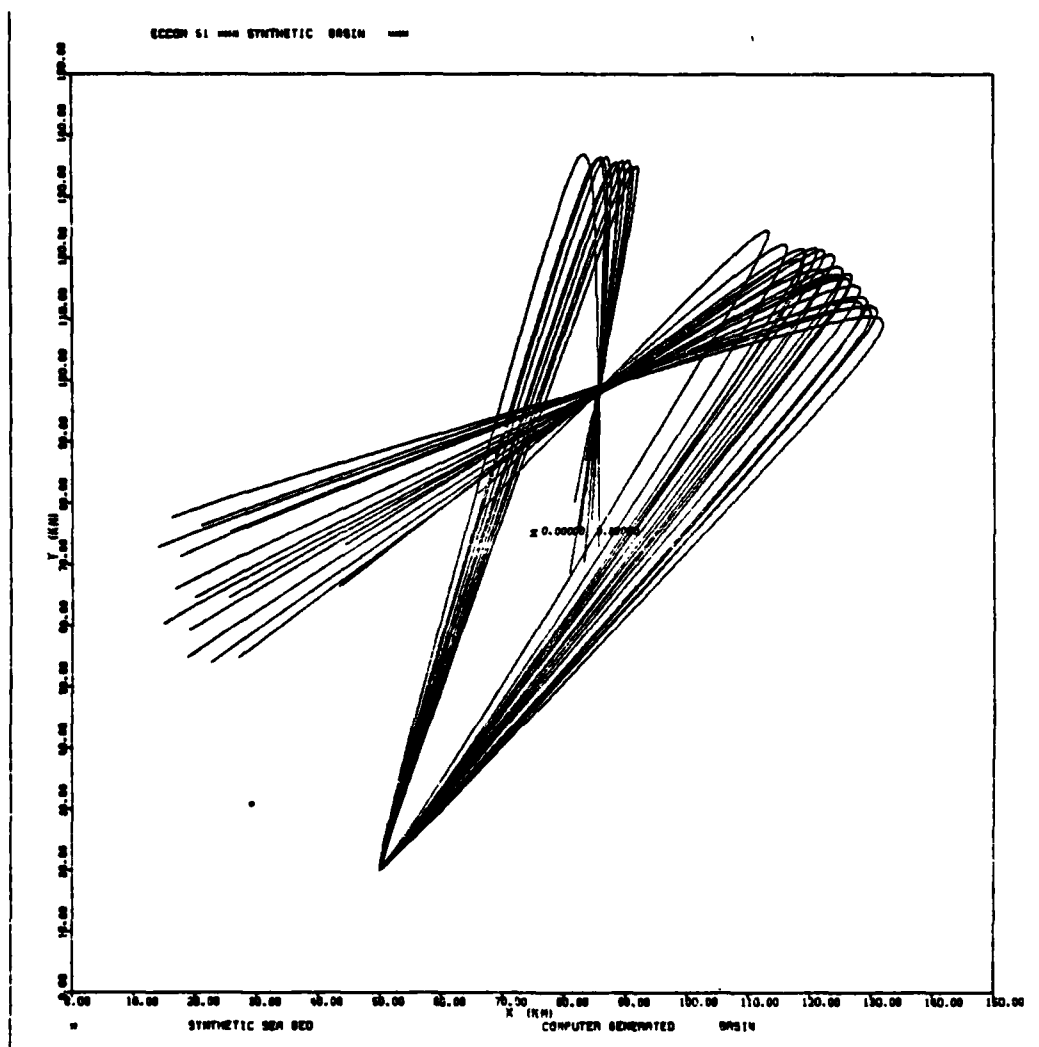
FIGURE 7. Convergence in a parabolic basin (Fig. 10). This is an
ECCOM product using punched output from three separate
ECTRACE runs. TABLE VII identifies the ray headings,
elevation angles, and number of points.
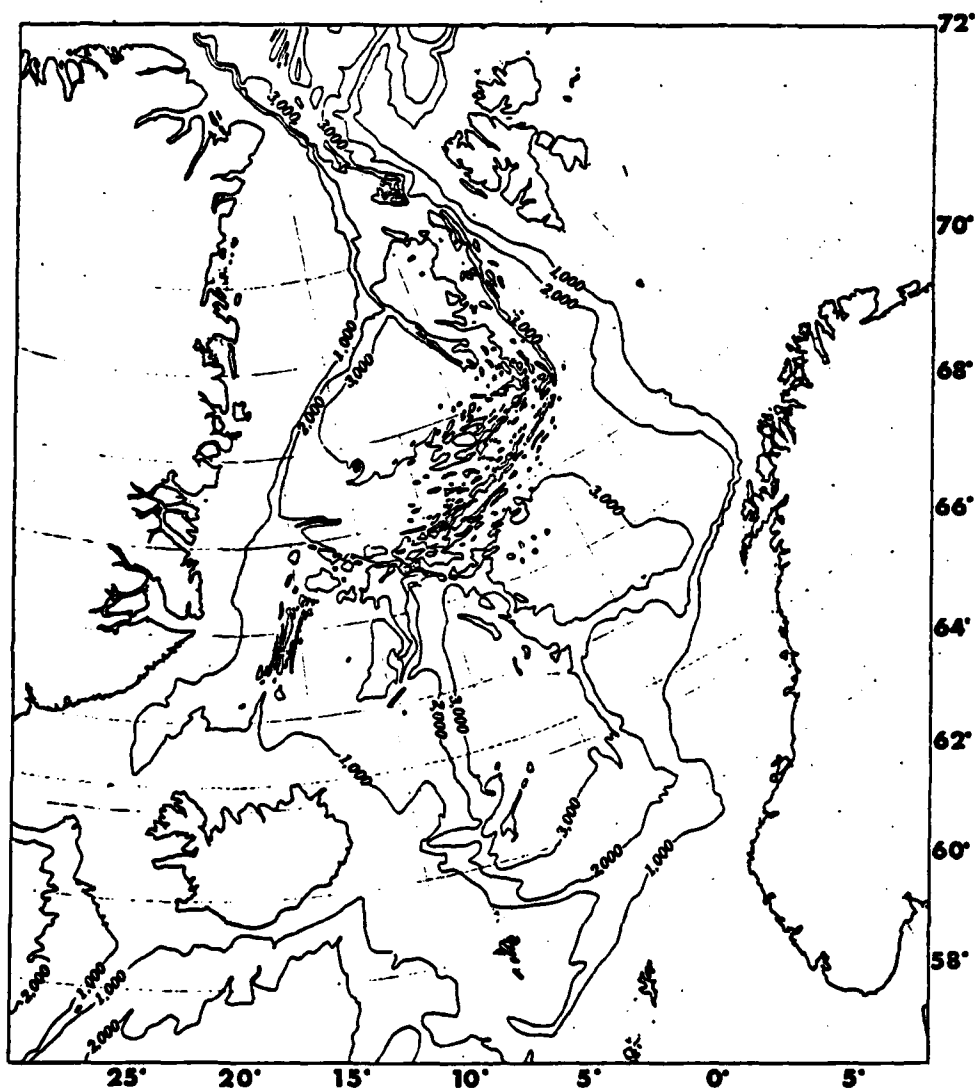
FIGURE 8. A reduced representation of a chart titled "Bathymetry of the Norwegian-Greenland and Western Barents Seas" published by NRL (Washington, D.C.). The actual chart measures 4 by 3 feet depicting finely grained contours. GENBOT uses bathymetry data files of this area to generate the matrix grids used by ECTRACE.
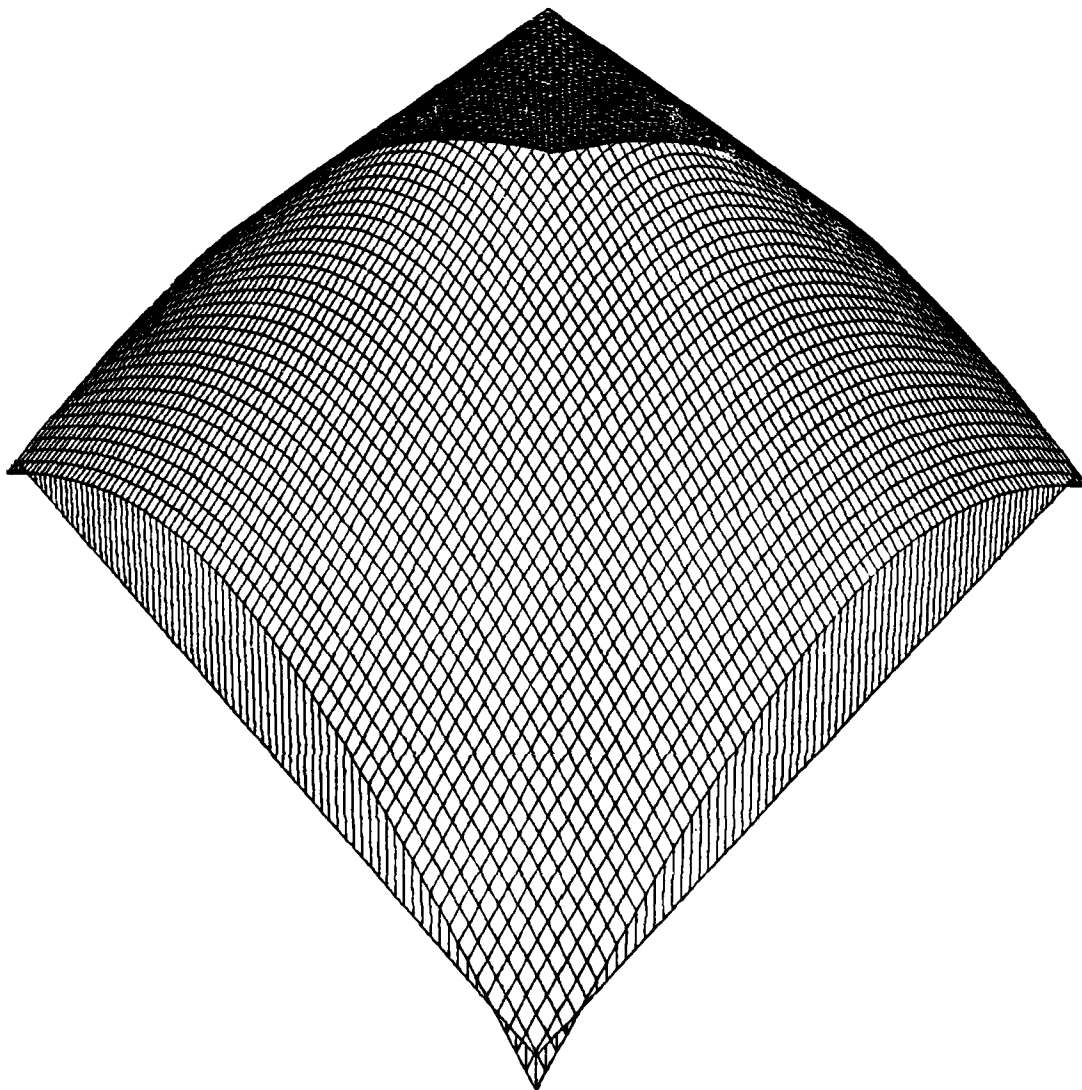
SYNTHETIC SEA BED
COMPUTER GENERATED   SEAMOUNT

FIGURE 9. A perspective surface plot produced by G3DP of a conical
          seamount generated by SYNGEN.
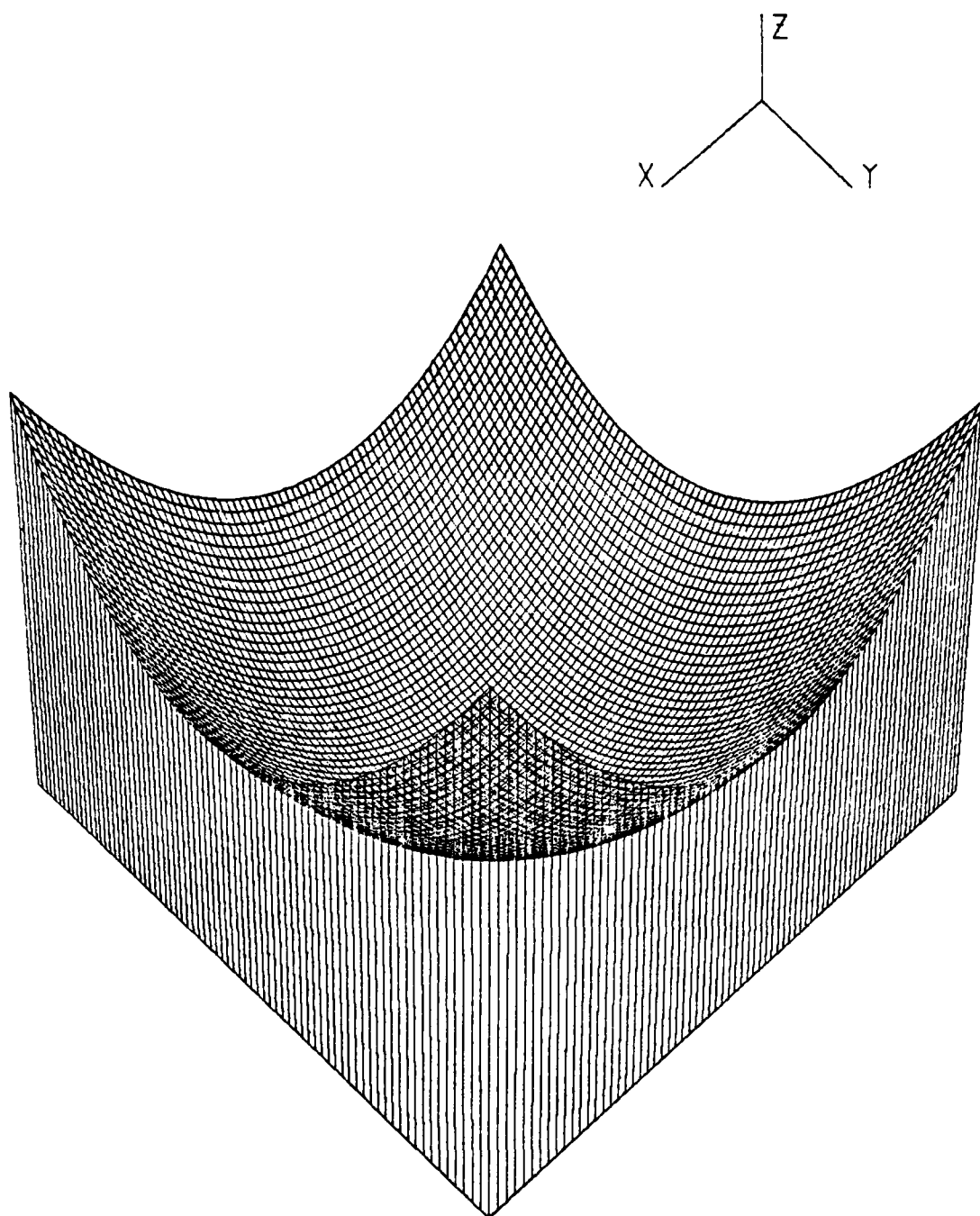
SYNTHETIC SEA BED
COMPUTER GENERATED   BASIN

FIGURE 10. A perspective surface plot of a synthetic parabolic basin
produced by SYNGEN.

119
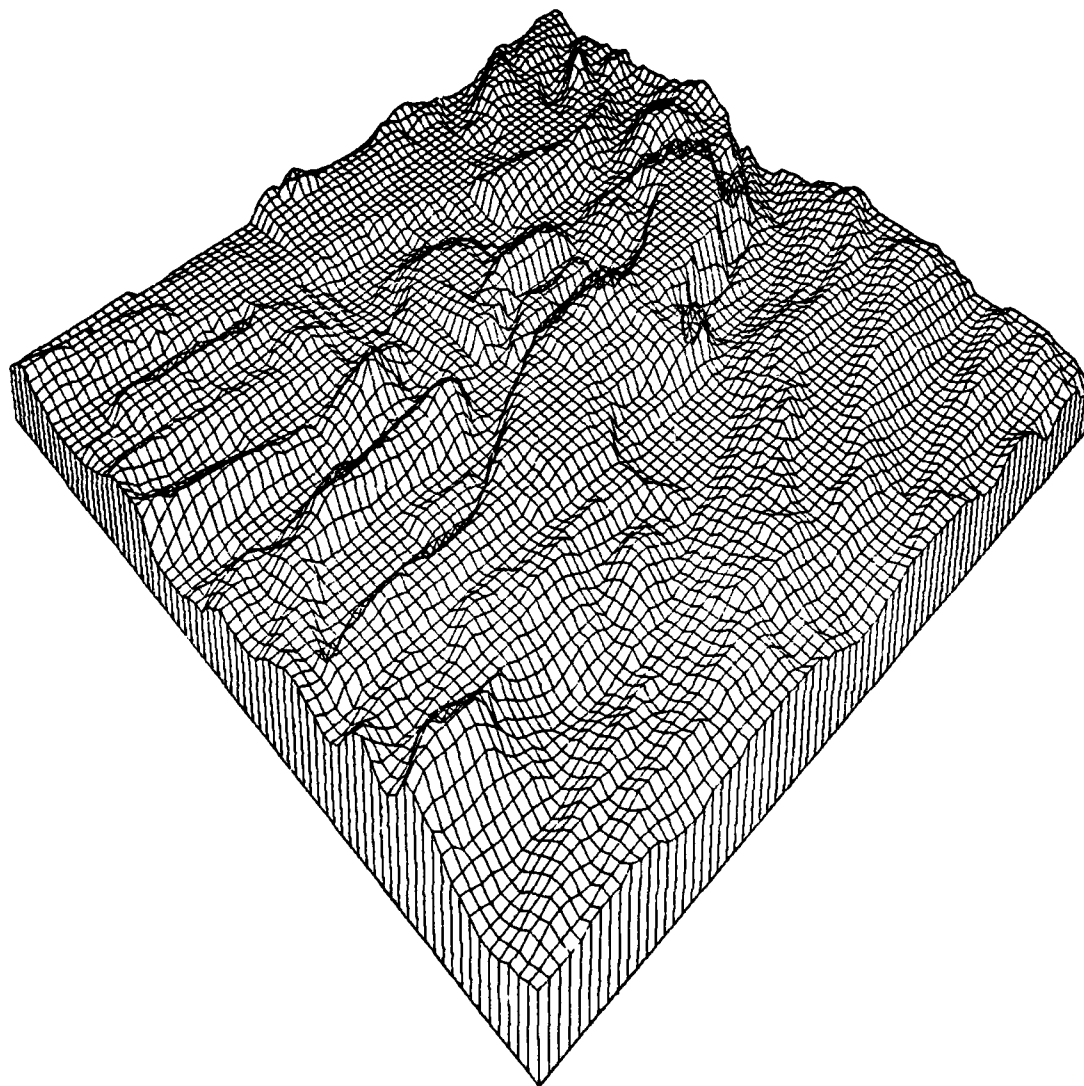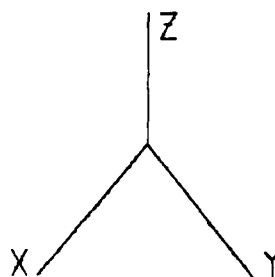
MØHNS RIDGE
CENTER LAT/LØN 71-37N/000-02W

FIGURE 11. A perspective surface plot produced by G3DP of a depth
matrix representing a portion of the Mohns Ridge, see
Fig. 2.
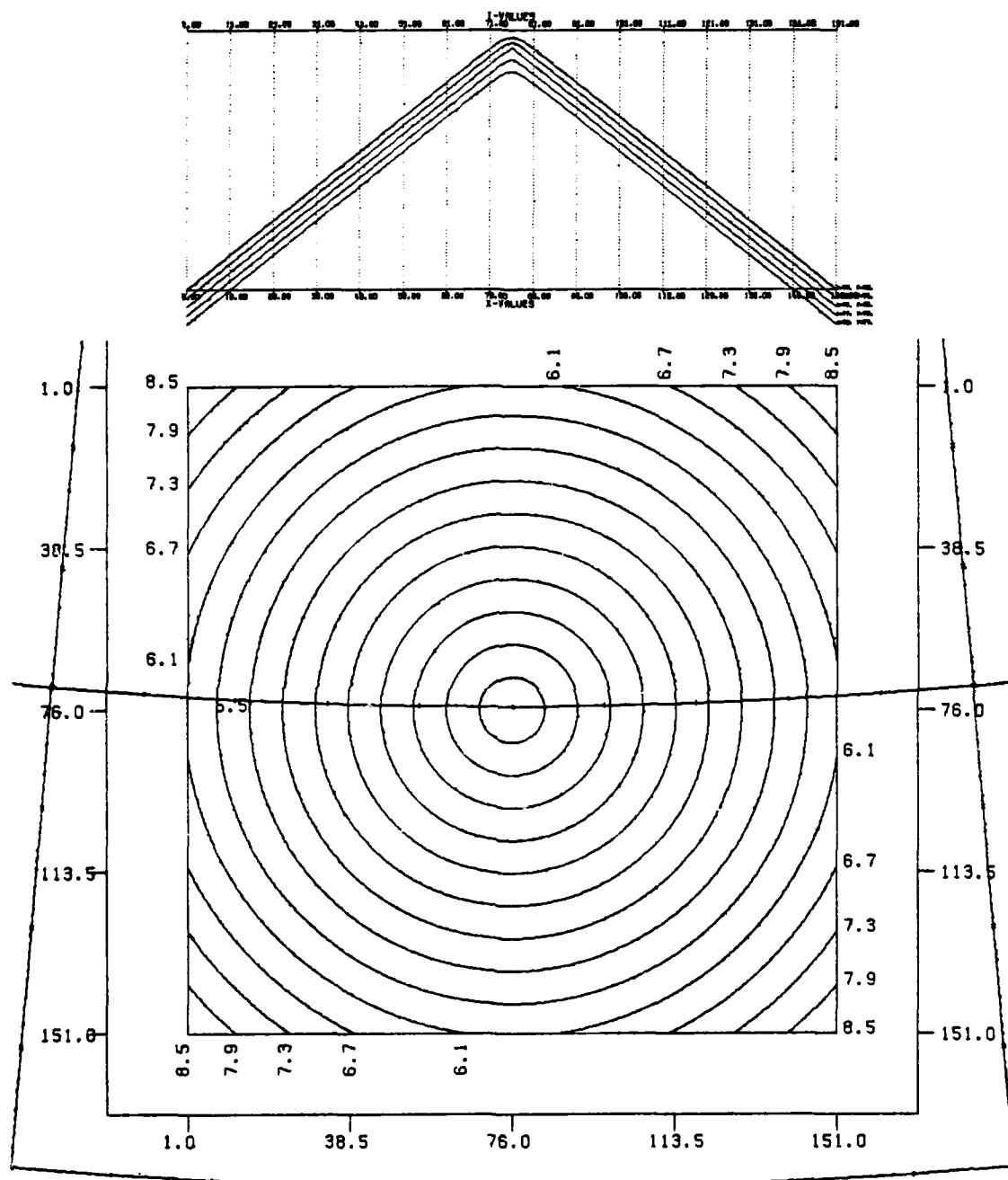
SYNTHETIC SEA BED
COMPUTER GENERATED   SEAMOUNT

FIGURE 12.   A graphical analysis performed by GRDCHK of a depth matrix
             representing a conical seamount. The upper portion is a
             series of displaced plots of selected columns of the matrix.
             The lower portion is a contour plot. The seamount was
             created by SYNGEN, see Fig. 9.

121
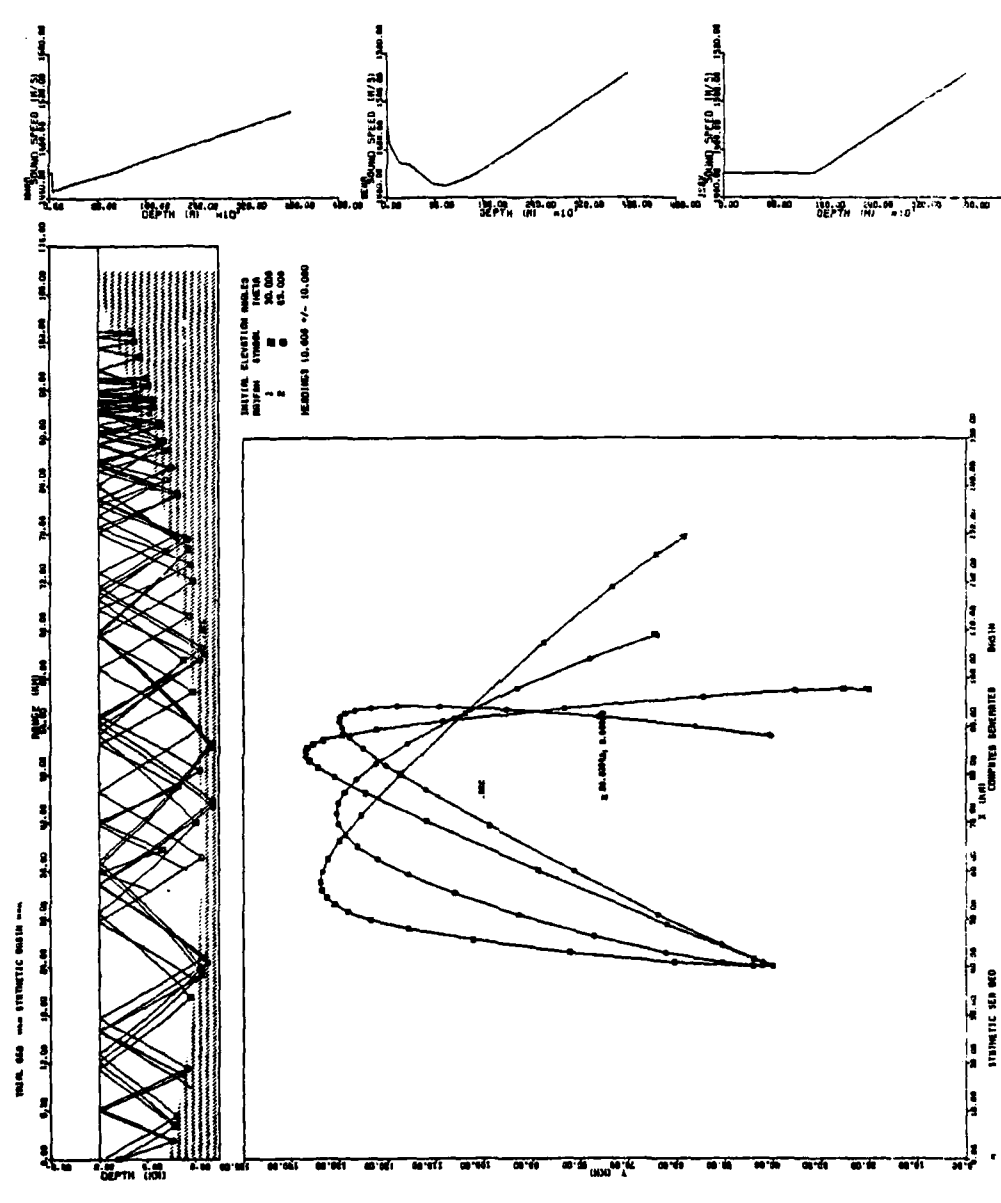
FIGURE 13. An ECTRACE plot of two ray fans in a synthetic parabolic
basin (FIGURE 10) generated by SYNGEN. The ray fans have
initial elevation angle of 30 and 45 degrees (positive
angle downward). Each ray fan contains two rays with
initial headings 360 and 020 degrees, respectively. The
plot was terminated after each ray had accumulated 23
bottom bounces.
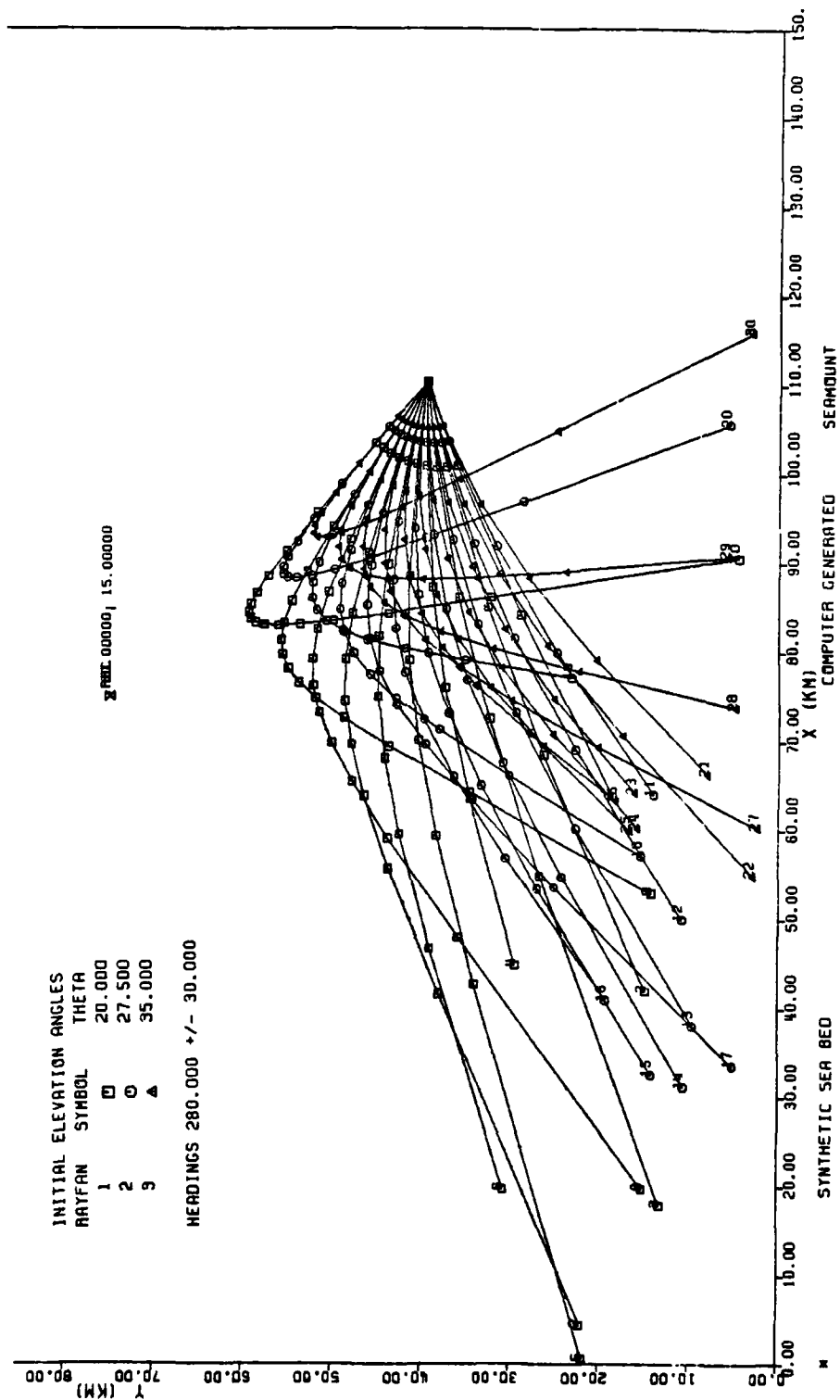
FIGURE 14. An ECTRACE horizontal product of three ray fans contacting a synthetic
seamount, see Figs. 9 and 12.

TRIAL 54 *** SYNTHETIC BASIN ***
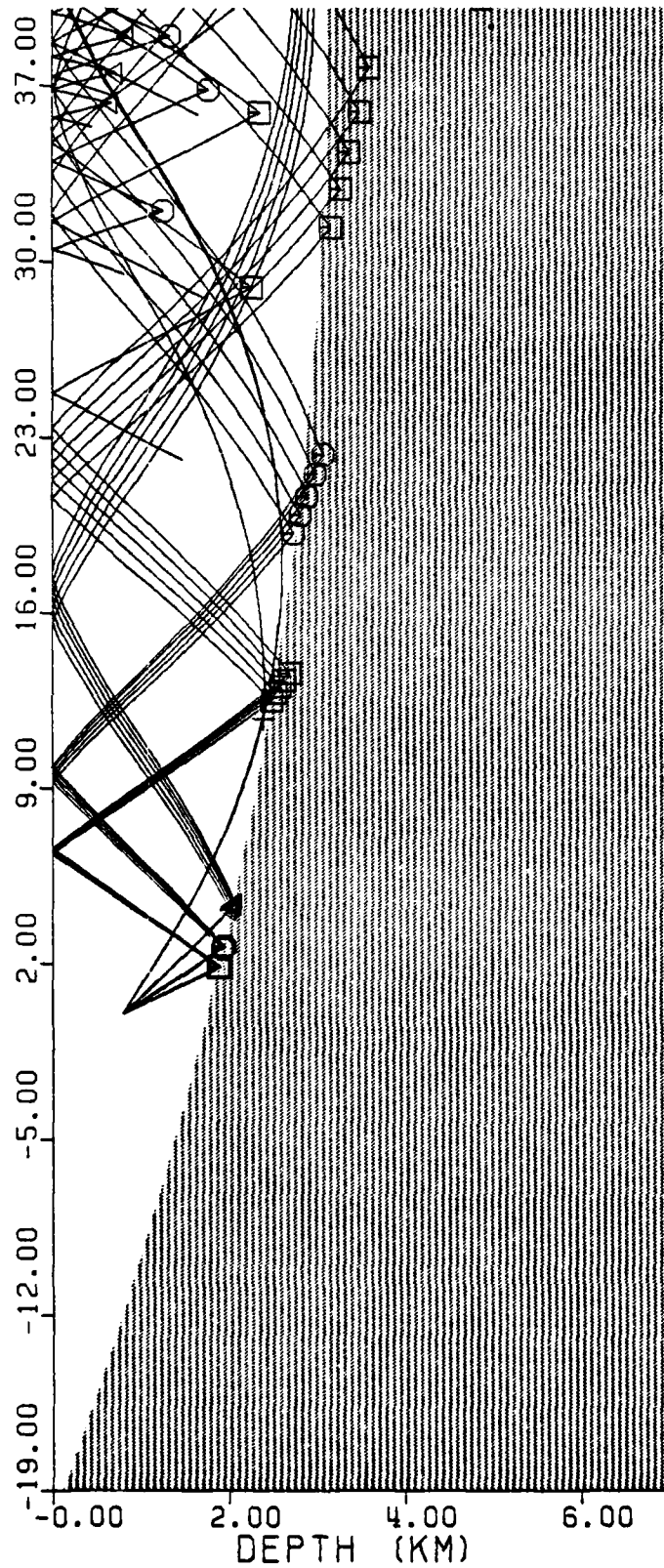
FIGURE 15. ECTRACE TRIAL, 54. There are 3 ray fans of 5 rays per fan. The ray fan depicted with the square symbol, can be seen first to curve clockwise away from the bathymetry slice, and then reverse and curve counterclockwise. Eventually, the rays can be seen to appear to approach the source. Since the bottom profile is taken only along the mean heading of the ray fan, most bounce symbols will not lie on the profile surface.
(FIGURE 15 continued)

124
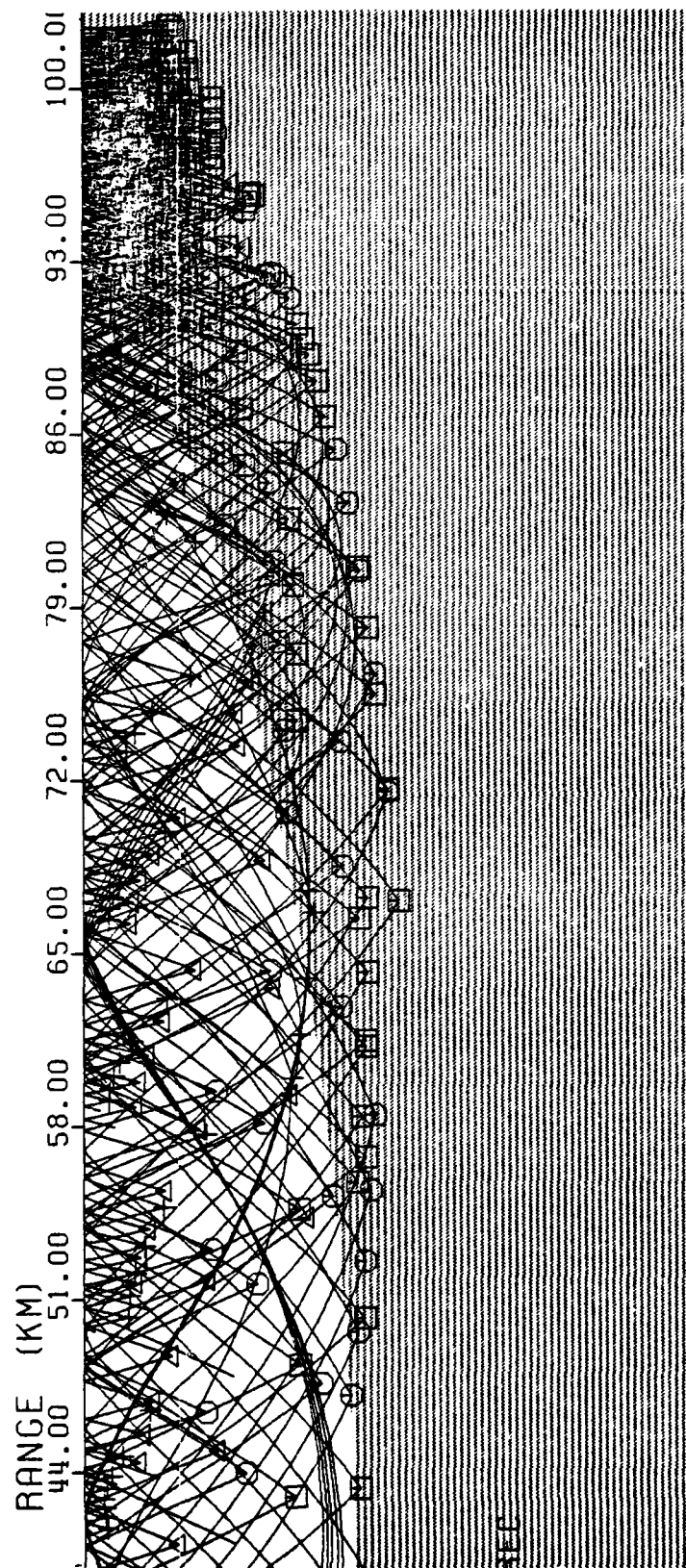
continued FIGURE 15.

TRIAL 31 *** SYNTHETIC SEAMOUNT ***
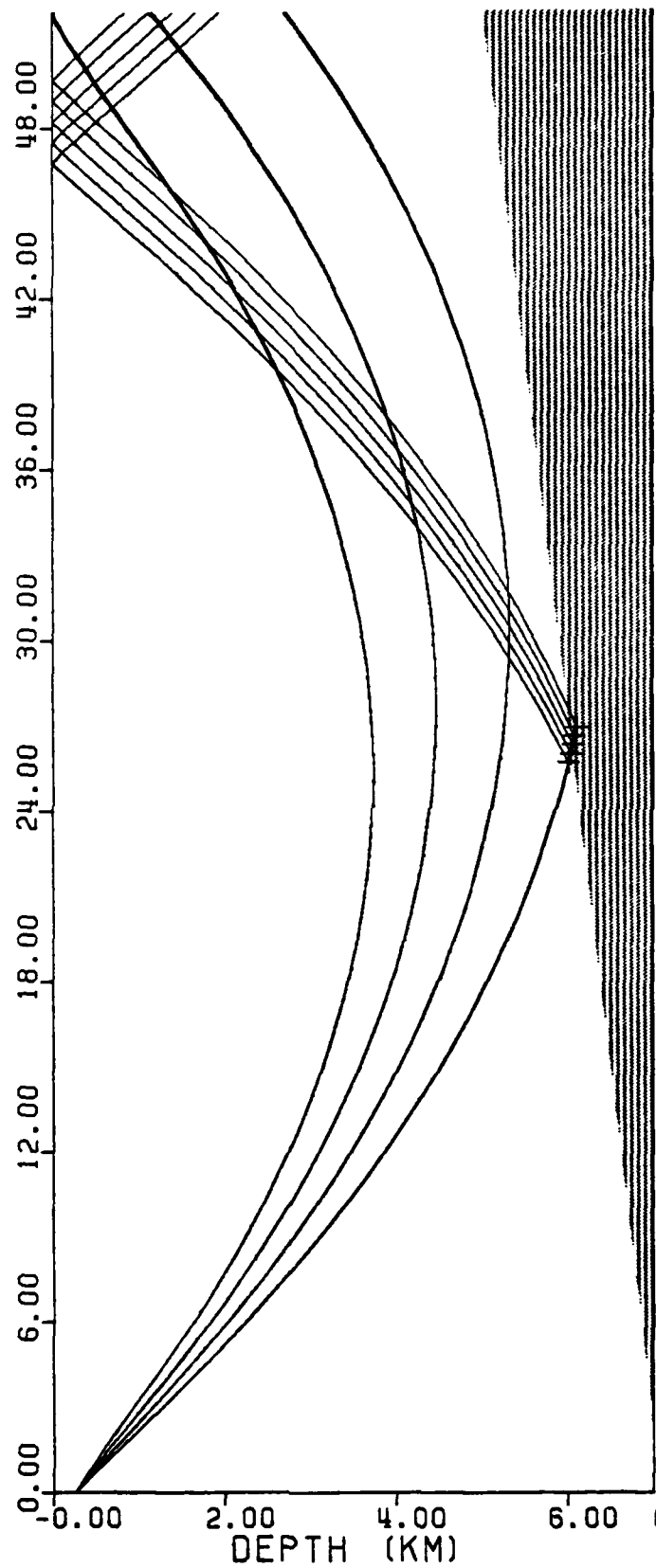
FIGURE 16. The vertical plot of an ECTRACE run over a synthetic grid representing a seamount. The axis of the ray fan was aimed at the center of the seamount. Rays in a single narrow ray fan are nearly indistinguishable until the first bottom contact is made.
(FIGURE 16 continued)

continued FIGURE 16

127

$\bar{x}$ 89.32843, 0.00000

40.00   50.00   60.00   70.00   80.00   90.00   100.00   110.00   120.00   130.00
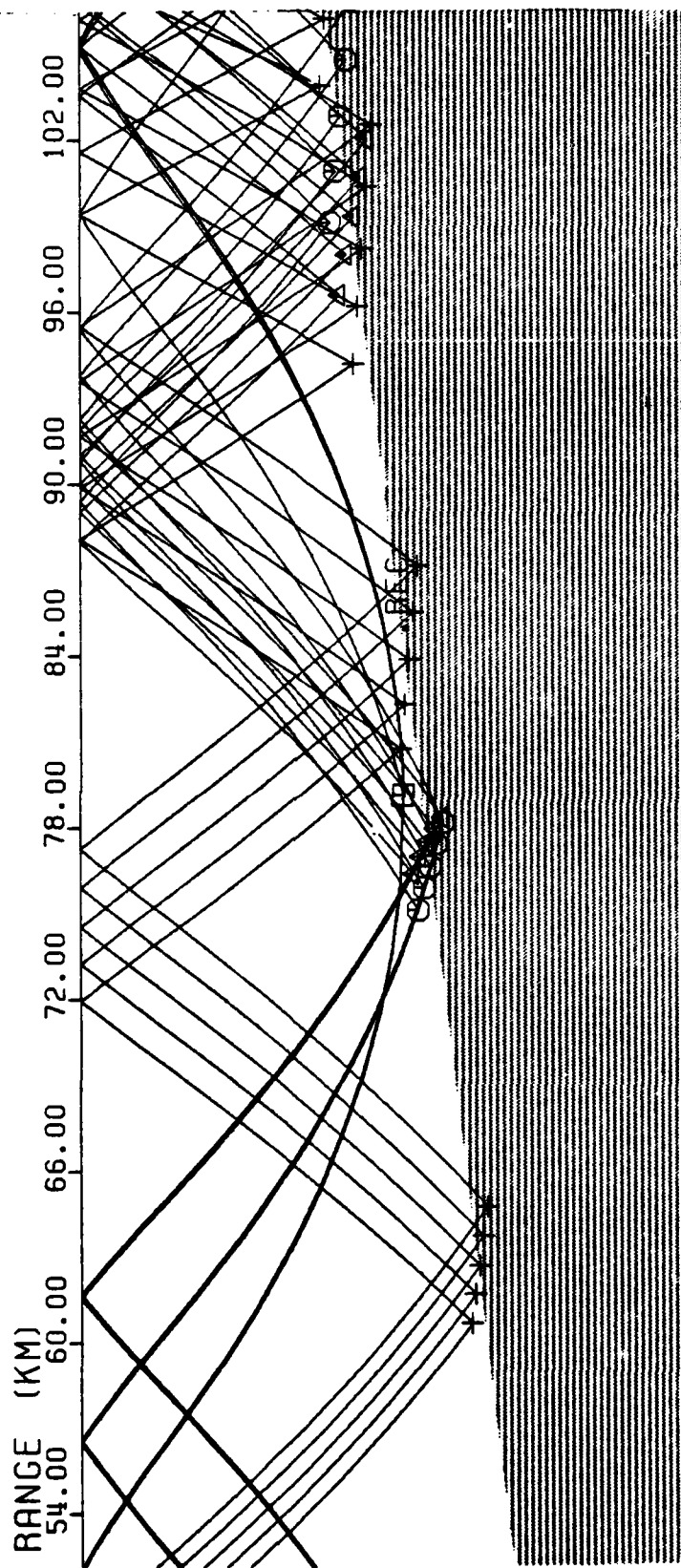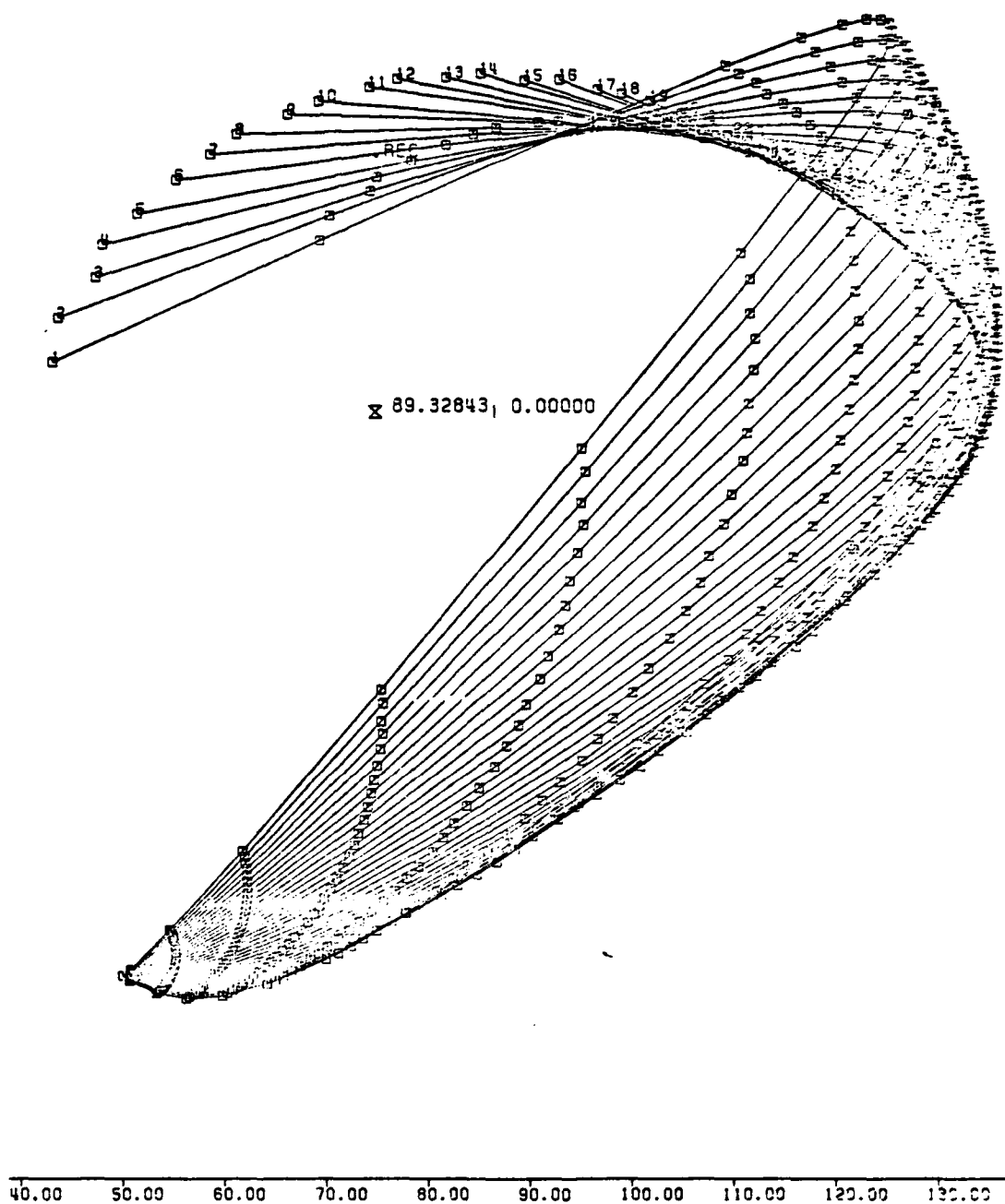
FIGURE 17.   An ECTRACE horizontal plot product of a single wide ray fan
             in a synthetic basin. All rays share the same bottom con-
             tact symbol. The number at the last symbol identifies the
             ray's order in the printed and punched history.
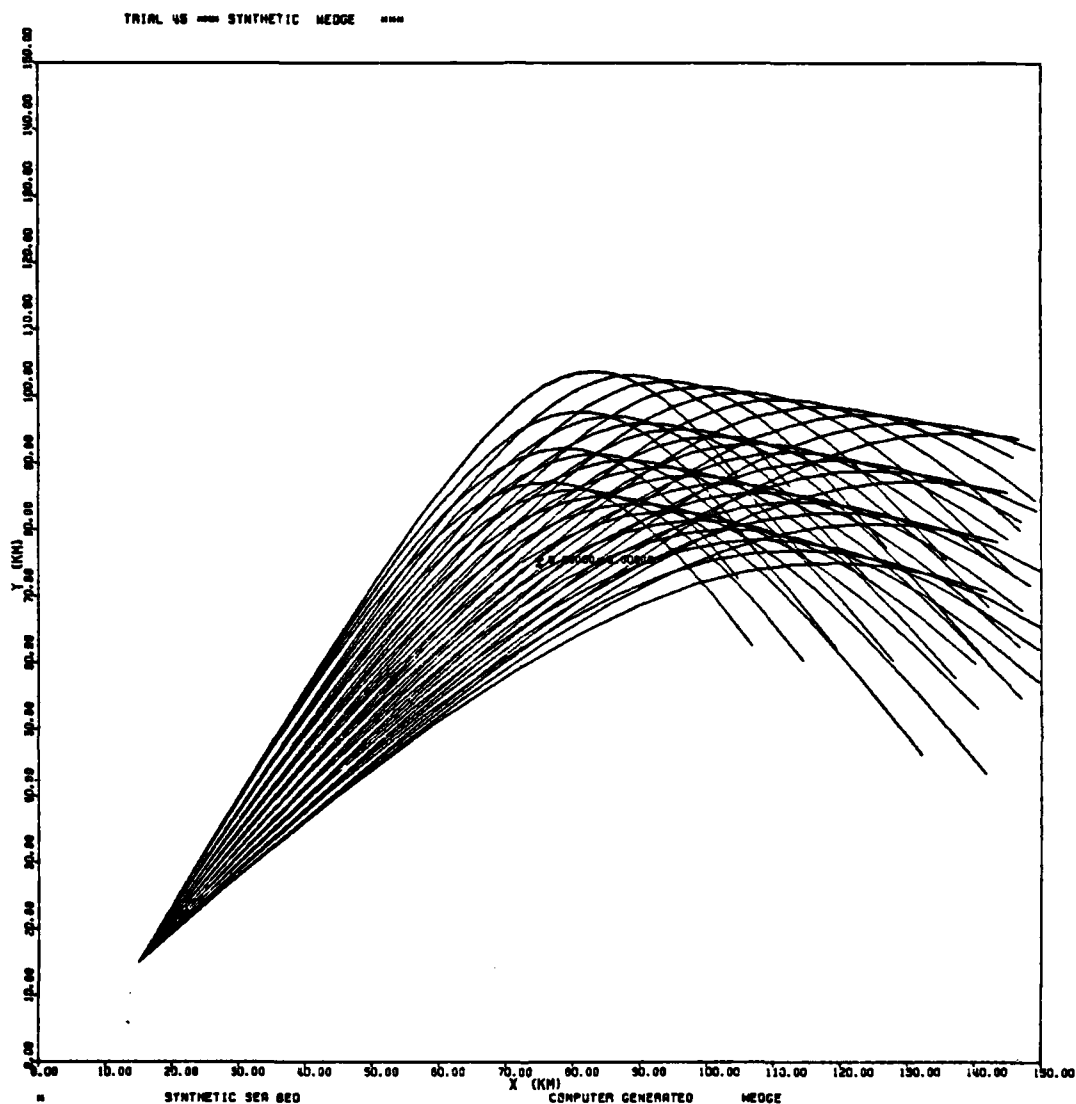
FIGURE 18. A horizontal plot produced by ECCOM of four ray fans propa-
gating over a six degree wedge. The punched output of three
ECTRACE runs was combined and then every other ray data his-
tory card sequence was removed. There are a total of 40 rays
depicted varying in initial elevation angles from 20 to 26
degrees in two degree increments. The initial headings vary
from 31 to 49 in two degree increments. The main purpose of
ECCOM is to display the horizontal curvature effect of a
large number of rays.

```
*******************************************************

                        GENBOT

*******************************************************

//GENBOT  JOB  (2648,0354,IX82),'ERAZO',TIME=20
// EXEC  FORTCLCG,REGICN.CC=250K
//FCRT.SYSIN CC *
C   PROGRAM:      GENBOT
C   SOURCE/CATE:  L. R. ERAZO, DEC 75
C   PURPCSE:      READS BATHYMETRY CONTOURS FROM MOBIL CIL CO.
C   DATA FILE AND GENERATES A SQUARE MATRIX CF DEPTH VALUES LYING
C   IN A DESIGNATED REGION OR PRODUCES A CONTCUR PLCT FROM CATA.
C
C   SUBRCUTINE GENDST MAY EE FCUND IN CRCCHK APPENDIX LISTING
CC
      IMPLICIT REAL*8(B-H),REAL*8(C-W)
      CIMENSICN TTL(12)
      DIMENSICN AX(5000),AY(5000),AZ(5000),AT(5000),AZB(10)
      DIMENSICN NAME(8),GLAT(4),GLCN(4)
      INTEGER*2 KEYX(5000),KEYY(5000)
      LOGICAL LEAD,LFLCT,LSTF,LGRIC
      CCMMCN /FLCTIT/ TTL,CNX,CNY,CNLAT,CNLON
      CCMMCN /CHART/ XMIN,XMAX,YMIN,YMAX,XRG,YRG,APDV
      CATA NAMET/'NAME'/
      LINT(CFARG) = AINT(SNGL(CPARG))
      CNE=1.
      PI=CATAN(CNE)*4
      CETRA=PI/180.
      KFIT=C
      LSTP=.FALSE.
      IFILE=0
C   USER INPUT GRID FARAMETERS: TITLE, LAT/LCNG OF CENTER,
C   SFACING IN KM (RESCLUTICN) CF DEPTH VALUES, RADIUS
C   CF AREA, TECHC CA CRIC FILE.
      READ 5000,TTL,CNLAT,CNLCN,DHN,ARAD
```

130

```
500C  FORMAT(6A8/4F10.5)
501C  READ 501C,AFILES,LEN
      FORMAT(16I5)
      LGRID=LEN.GT.0
      LEN=JAEV(LEN)
      LFLCT=LEN.GE.4
      IF(.NOT.LGRID) GO TO 1
6000  PRINT 6000,TTL,CNLAT,CNLCN,NFILES
      FORMAT(10X,'GRID FILE CALLEC ',;2(T40,6A8/),10X,'WILL BE CONSTRUCTE
     1C ARCLNC LAT=',F10.5,' LCN=',;CABS(CNLAT))
      ARADMX=111.12*(85.-CABS(CNLAT))
      ARAC=APIA11(ARAD,ARACMX)
      IF(ARAC.LT.1.5) GO TO 5CC
    1 IMAX=IFIX(2*SNGL(ARAC/CHA))+1
      JMAX=IMAX
      XMAX=(IMAX-1)*DHN
      YMAX=XMAX
      XMIN=C.
      YMIN=C.
      XRG=XMAX
      YRG=YMAX
      APCV=APMAX1(XRG,YRG)/15.
      CNX=ARAC
      CNY=ARAC
      CALL CCFMER(CLATMN,CLATMX,GLONMN,GLONMX,IMAX,JMAX)
C     FLCT GEOGRAPHIC ECLCARY LINES.
      CLT=1.
      CLTC=.25
      ABL=CMAX1(CABS(GLATMX),CABS(GLATMN))
      CLA=CMAX1(CABS(GLATMX),CABS(GLATMN))
      IF(AEL.LT.85.) DLN=CLN/2
      IF(AEL.LT.6C.) DLN=CLN/2
      CLLNC=1.
      IF(AEL.GT.75.) GO TO 310
      CLLA=1.
31C   CLLAD=CLTC
      ELCN=GLCAPA-DLN
      ELCN=GLCAPX+CLN
      CLAT=GLATPA-CLTT
      ELAT=GLATPX+DLTT
      IF(SLCR.LT.CNLCN-9C.) SLCN=SLCN+DLN
      IF(ELCN.GT.CNLCN+9C.) ELCN=ELCN+DLN
      SLAT=CMAX1(SLAT,DBLE(-85.))
      ELAT=CMAX1(ELAT,CBLE(+85.))
      IF(.NOT.LFLCT) GO TO 5
      AFACIZE=LEN/4*21.
      ASIZE=ZFAC(4)*21.
      CALL HINCCL(C.,ASIZE,C.,ASIZE)
```

131

```
      CALL PLCTIS(2,2,2)
      CALL FACTCR(AFACT)
      CALL PLCTT(C..99.,2C..5..2)
      CALL PLCTT(2C..99.,2C..5..2)
      CALL PLCTT(2C..0..2)
      CALL PLCTT(0..0..2)
      CALL PLCTSYMBCL(-2.4,-2.4,.12,TTL,0.,96)
C     DRAW LATITUDE LINE
      GJ=CCNT(SLAT)
      GJ1=CINT(SLCN)
      IF(CLCN.AE.1.) GJ1=1C*CINT(GJ1/10.)
      IF(GJ1.CE.GJ2) GC TC 5C
      CALL CEFLT(GI,GJ1,GJ2,D.,AX,AY,1)
      GI=GI+CLT
      IF(GI.LLE.ELAT) GC TC 340
  340 CRAW LCNGITUDE LINE
C
      GJ=CLTC
      GJ1=CINT(SLAT)
      GJ2=ELAT
      CALL CEFLT(GI,GJ1,GJ2,D.,AX,AY,2)
      GI=GI+ELCN) GC TC 250 SNGL(DLTD),0.,2)
      CALL NUMBER(13.,-2.4,.1,'1.'DEG PER +',.0.,12)
      CALL PLCTPEER(.999.,.999.,1.'SNGL(DLND),0.,2)
      CALL PLCTPEER(.999.,.999.,1.'DEG PER X',.0.,10)
  250 BEG IN A NEW FILE.
C
      IFILE=IFILE+1
      ICCN=0
      NPTF=0
      READ(1,15C10) LATCR,LATCF,NCON
      LATE=LATCR+1C
      LCAE=LCACF+1C
 15CC PRINT 1SCC,IFILE,LATCR,LATE,LCNCR,LONE,NCCN
      FORMAT(//1CX,'FILE ',I2,' CCVERS LAT ',I3,' TO ',I3,', LCN ',
     1I3,' TC ',I3,' WITH ',I2,' CCNTCUR LINES.',//T9,'CCNTOUR LINE',T23
     2,'CEFTH',T36,'LABEL',T47,'NPTS',T57,'KPTS')
C     THE NEXT RECCRD IS EITHER A LINE LABEL CR THE HEADER RECORC
C     CF A NEW CCNTOUR LINE
   43 READ(1,1400,END=100) (NAME(I),I=1,3),CLAT,CLON,INDEP,METER
 14CC FORMAT(A3,A4,A4,7X,2FIC.5,2X,I4,1X,A2)
      IF(NAME(2).EC.NAME(1)) GO TO 85
```

132

```
C     PROCESS A NEW CONTOUR LINE.
      IPLTST=0
      KPTS=0
      NPTS=0
      LEADR=.FALSE.
      ICCA=ICCA+1
      CDC=INDEF/IC.
C     BEGIN READING LATS/LONS ALONG CONTOUR LINE.  EACH RECORD
C     HAS FOUR LAT/LON PAIRS.
      READ(1,1500) (GLAT(I),GLON(I),I=1,4)
1500  FORMAT(8F10.5)
      KE=4
50    DO 50 K=1,KE
      LEADR=(GLAT(K).GT.90..OR.GLAT(K).LT.-90.
      IF(LEADR) GO TO 60
      CONTINUE
      GO TO
C     MORE LINE DATA FOLLOWS.  BUT FIRST:
C     END OF CONTOUR LINE.  SET KE TO NUMBER OF POINTS IN RECORD.
      KE=K-1
      IF(KE.LT.1) GO TO 60
C     CHECK RELEVANCE OF POINTS.
65    DO 70 K=1,KE
      IF(GLAT(K).LT.SLAT.OR.GLAT(K).GT.ELAT) GO TO 70
      IF(GLON(K).LT.SLON.OR.GLON(K).GT.ELON) GO TO 70
C     FIND POINT'S POSITION ON GRID.
      CALL GECOST(CNLAT,CNLON,GLAT(K),GLON(K),HS,HF,DIST,-1)
      KPTS=KPTS+1
      KPTT=KPTT+1
      IF(KPTT.LT.4998) GO TO 66
      PRINT 655,ICX,
655   FORMAT(10X,'*** DIMENSION LIMIT REACHED ***')
      LEADR=.TRUE.
66    IF(IPLTST.EQ.0) IPLTST=KPTT
      IPLTEN=KPTT
      AY(IFLTEN)=DIST*DCOS(HS)+CNY
      AX(IFLTEN)=DIST*DSIN(HS)+CNX
      AZ(IFLTEN)=CDD
70    CONTINUE
      NPTS=NPTS+KE
      IF(.NOT.LEADR) GO TO 45
80    PRINT 1000,ICON,CCC,NAME(1),NAME(2),NAME(3),NPTS,KPTS
1000  FORMAT(10X,F10.2,1X,A3,2A4,I8,I10)
      NPTF=NPTF+NPTS
      IF (KPTS.LE.2) GO TO 40
      KPTF=KPTF+KPTS
      IF((.NOT.LFLCT.OR.KPTS.LE.1).AND..NOT.LSTP) GO TO 40
```

133

```
C        PLCT CCNTCLR LINE.
         KPFF1=KFFTT+1
         KPFF2=KFFTT+2
         AX(KPFFF1)=XMIN
         AY(KPFFF1)=YMIN
         AX(KPFFF2)=XFCV
         AY(KPFFF2)=AFCV
         CALL LINE(AX(IPLTST),AY(IPLTST),KPTS,1,0,0)
         IF(.ACT.LSTP) GO TC 40
C        EAC CF CCNTOUR FILE.
  100    PRINT 1700,KFTF,NFTF,KPTT,ICCN
  17CC   FORMAT(1CX,'NUMBER CF BCTTCM FCINTS USEC=',I7,' (CUT OF ',I5,')',A
  1CCUM TCTAL =',I7//1CX,'NO. OF CONTOUR LINES FRCCESSED FRCM FILE=',I
  25)
         IF((IFILE.LT.NFILES).AND..NGT.LSTP) GC TC 9
C        SCRT CATA FOR GRIC CCNSTRUCTICN.
         IJ=0
         IERT=0
         CALL RAIN1(KPTT,AX,AY,AT,KEYX,KEYY,IER)
         X1=AX(KEYX(1))+30.
         Y1=AY(KEYY(1))+30.
         X2=AX(KEYX(KPTT))-30.
         Y2=AY(KEYY(KPTT))-30.
         I=1
         IF(XMIN.GE.X1.AND.XMAX.LE.X2) GO TO 401
         IHAF=IMAX/2
  400    I=2*IHAF
         XMIN=XMIA+CH-N
         XMAX=XMAX-CH-N
         IF(XMIN.LE.XMAX) GC TC 401
  401    CCATINUE
         IMAX=IMAX-2*(I-1)
         CNX=CHAF*FLCAT(IMAX-1)/2.
         IF(YMIN.GE.Y1.AND.YMAX.LE.Y2) GO TO 411
         JHAF=JMAX/2
  410    J=2*JHAF
         YMIN=YMIN+CH-N
         YMAX=YMAX-CH-N
         IF(YMIN.LE.YMAX) GC TC 411
  411    CCATINUE
         JMAX=JMAX-2*(J-1)
         CNY=CHAF*FLCAT(JMAX-1)/2.
         IF(.ACT.LFLCT) GO TC 210
C        PLCT GRIC BOUNCARY LINES.
         XST=XMIN/APCV
```

```
      YST=YMIN/APDV
      XEND=XMAX/APDV
      YEND=YMAX/APDV
      XVAL=(X4+XMAX)/APDV
      CALL AXIS(XST,YST,'Y  (KM)',6,YVAL,90.,YMIN,APDV)
      CALL PLOT(XST,YST,2X)  (KM)',-6,XVAL,0.,XMIN,APDV)
      CALL AXIS(XST,YST,2X)
      CALL PLOT(XEND,YST,2)
      CALL PLOT(XEND,YEND,2)
      CALL PLOT(XST,YEND,2)
      CALL CORNER(C,C2,C2)
  210 CALL CORNER(CLATMX,GLONMX,IMAX,JMAX)
      ARAD=CAIA1(CAX,CNY)
      IF(ARAD.GE.7.5) GO TO 215
      PRINT(ESCC,ARAD)
 65CC FORMAT(10X,'SEMI-AXIS LENGTH OF ',F6.2,' KM TOO SMALL.'/10X,
     1'NO GRID WILL BE PRODUCED.')
      STCP
  215 IF(.NOT.LGRID) STOP
C     COMPUTE GRID CCNSTRUCTICN.
      WRITE(2,5030)TTL,CALAT,CALCN,CHN,IMAX,JMAX
      NREC=0
 5030 FORMAT(6A8/6A8/3F10.5,2I5)
      DO 200 J=1,JMAX
      AYJ=CLAY*(J-1)+YMIN
      DO 201 I=1,IMAX
      IJ=I+J+1
      AXI=CLAX*(I-1)+XMIN
      CALL ARAIN2(KFTT,AX,AY,AZ,KEYX,KEYY,AXI,AYJ,AZK,IER)
      IF(IER.EC.C) GO TO 2C2
      IERT=IERT+1
      PRINT(CC3C,  AXI,AYJ,I,J,AZK,IER
     1,IER)
 603C FORMAT(10X,'X=',F8.4,' Y=',F8.4,' ZB(',I3,',',I3,')=',F8.4,' IER='
     1,I5)
  202 IF(IERT.GE.1C00J) GO TO 205
      AZB(IJ)=AZK
      IF(IJ.LT.102) GO TO 201
      WRITE(2,1020) AZB
      NREC=NREC+1
      IJ=0
  201 CONTINUE
  200 CONTINUE
      IF(IJ.EC.C) GO TO 205
      WRITE(2,1020) (AZB(II),II=1,IJ)
      NREC=NREC+1
 1020 FORMAT(10(F8.4)
  205 PRINT (CCIC, NREC,IERT
```

135

```
6010  FORMAT(/1CX,'GRID CONSTRUCTION COMPLETED.',I5,' RECORDS IN FILE.',/
     110X,I5,' FCINTS IN QUESTION.')
      STCP
ENDFILE 2
C
C     DRAW LINE VALUES ON LINE.
   65 IF(.NCT.LFLCT)GO TO 40
      IF(DLAT.LT.SLAT.OR.CLAT.GT.ELAT) GO TO 40
      IF(DLCN.LT.SLCN.OR.CLCN.GT.ELON) GO TO 40
      CALL GECCST(CNLAT,CNLCN,CLAT,DLON,HS,HF,DIST,-1)
      AXS=(CDIVIGT*DSIN(HS)+CNX)/APDV
      AYS=(CDIVIGT*ECCS(HS)+CNY)/APDV
      CALL ECL(AXS,AYS,-1,METER,0.,2)
   40 GO TO
      END
      SUBROUTINE CORNER(CLATMX,CLATMN,CLONMX,CLONMN,IMAX,JMAX)
      IMPLICIT REAL*8(A-H),REAL*8(O-W)
      DIMENSION TTL(12)
      COMMCN /FLCTIT/ TTL,CAX,CAY,CNLAT,CNLCN
      COMMCN /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APDV
      DIST=DSCRT(CNX**2+CNY**2)
C     CALCULATE LAT/LCN OF GRID CORNERS. C1 IS SW AND C2 IS SE.
      1S=CATAN2(CNX,CNY)
      CALL GECCST(CNLAT,CNLCN,C3Y,C3X,HS,HF,DIST,1)
      1S=CATAN2(-CNX,-CNY)
      CALL GECCST(CNLAT,CNLCN,C2Y,C2X,HS,HF,DIST,1)
      C4Y=C3Y-CNLCN
      C4X=CNLCN-CNLCN
      CLCA=CALCN-CNLCN
      CLCA=CALCN-CNLCA
      C4X=CALCN-CNLCA
      C1X=CALCN-CNLCA
C
C     CALCULATE BOUNDARY LATS/LONS THAT FORM SPHERICAL RECTANGLE
      WHICH INCLUDES GRID.
      (E.0.) GC TO 5
      IF(CNLAT.GE.CNY) CIX
      CLATPA=CNY
      CLCNPA=CNX
      CLCNPA=CNX
      CALL GECCST(CNLAT,CNLCN,CLATMN,CNLON,180.,HFC,CNY,2)
      CC
      CLATPA=C4Y
      CLCNPA=C4X
      CALL GECCST(CNLAT,CNLCN,CLATMX,CNLCN,CLATMX,CNLON,O.,HF,CAY,1)
    5 PRINT 6010,CNX,XMIN,YMIN,C1Y,C1X,C2X,C2Y,C2X,XMAX,YMAX,
     1 CLONMN,GLONMX,CLATMX,CLATMN,IMAX,JMAX
 6010 FORMAT(/Ti,'VALUES AT GRID CORNERS:'/T21,'X',T31,'Y',T4C
     1 'LAT',T50,'LCN'/4(17X,4F10.5/)/T14,'MIN LON',3X,'MAX LON',
     2 3X,'MIN LAT',3X,'MAX LAT',3X,'IMAX',3X,'JMAX'/T11,4F10.5,2I7/)
```

```
      RETURN
      END
      SUBROUTINE GEOPLT(GI,GJ1,GJ2,DJ,X,Y,LLL)
      IMPLICIT REAL*8(C-H),REAL*8(C-W)
      DIMENSION X(183),Y(183),TTL(12)
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLCN
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APCV
      INTEG=LLL+2
      IF(LLL.NE.1) GO TO 12
C     LATITUDE LINE GI FROM LONGITUDE GJ1 TO GJ2
      PLAT=GI
      FLCN=GJ1
      CLLAT=GI
      CLLCN=GJ2
      CLLAT=0.
      GO TO 20
C     LONGITUDE LINE GI FROM LATITUDE GJ1 TO GJ2
10    PLAT=GJ1
      FLCN=GI
      CLLCN=GI
      CLLAT=GJ2
      CLLCN=0.
20    CALL GEOCST(CNLAT,CNLCN,PLAT,PLON,HS,HF,DIST,-1)
      IF(FLAT.GT.CLAT.OR.FLCN.GT.CLCN) GO TO 30
      NPTS=NPTS+1
      X(NPTS)=DIST*DSIN(HS)+CNX
      Y(NPTS)=DIST*DCOS(HS)+CNY
      PLAT=PLAT+FLAT
      FLCN=FLCN+CLLCN
      GO TO 20
30    NPTS=NPTS+1
      APF1=NPTS+1
      APF2=NPTS+2
      X(NPF1)=XMIN
      Y(NPF1)=AFLCV
      X(NPF2)=APCV
      Y(NPF2)=APCV
      CALL LINE(X,Y,NPTS,1,1,INTEG)
      RETURN
      END
      SUBROUTINE GEOCST(SLAT,SLON,FLAT,FLON,AZ1,AZ2,DKM,MODE)
      IMPLICIT REAL*8(A-H),REAL*8(C-Z)
      LOGICAL LCEC
      DATA EFS/.0818193564/,ERAVO/6356.75023/,RIME21/2./
      FUNK(EFS,F)=DTAN(DATAN(EFS*SFSI/RIME21/2.)**2
      CNE=1.
```

```
      HAFI=CATAN(CKE)*2
      PI=HAPI*2
      TUPI=2*PI
      CETRA=PI/1.EC.
      EPS2=EFC/(CNE-EPS2)
      RIME2=EFC/*CET(CNE-EPS2)
      IF(SLAT.EC.CETRA.SLAT=1.E-5
      FLII1=SLAT*CETRA
      XLA1=SLCN/*CETRA
      IF(CABS(SLAT).LT.SC.)GC TO 1
      TPSI1=DSJGN(CELE(5.EC),SLAT)
      GO TC
    1 TPSI1=DTAN(FLII1)*RIME2
    2 FSII1=CATAN(TPSII).
      LCEG=IABS(PCCE).NE.1
      CKM=CAES(TEK2)
      IF(PCCE)2C0,100,100
C     FCRWARD SCLUTICN.
  10C ALFA1=AZI
      IF(LCEG) ALFA1=AZI*CETRA
      IF(ALFA1.EC.TUPT)ALFA1=C.
      IF(ALFA1.LT.O.)ALFA1=ALFA1+TUPI
      IF(ALFA1.LE.FI)GC TC 1C1
      CKM=CKM
      ALFA1=4.*FI-FI
  101 SALFA1=CSIN(ALFA1)
      CPSI1=CCCS(PSI1)*SALFA1
      CALFA1=CATAN2(-CALFA1,TFSI1)
      SIGC1=CATAN2(-CALFA1,TFSII)
      XLEI=CCSIN(CARCCS(CFSIN(PSI1)*SALFA1))
      SPSI1=CCSIN(CFSIM)
      XK=FLNK(/CFSIM)
      TWSIG1=TWSIG1+XK*(DSIN(TWSIG1)+XK*(DSIN(TWSIG1)-XK*(DSIN(2*TWSIG1)/8.))
      CSF=(1.-XK)*CKM/BRAVC
      TWSF=TWSF+CSF
      CSIC=CSF+XK*(XK*5*DSIN(2*DSP)*DCOS(2*TWSP)/8.-DCOS(TWSP)*DSIN(CSF)
     1)
      SIG2=SIG1+DSIG1
      SIG=(2*TWSIG1+DSIG1/2.
      XLE2=CATAN2(DSIN(SIG2),CCOS(SIG2)
      CLE8=XLB1-XLBI
      SPSI2=CCCS(SIG2)*SPSIM
      FSI2=CRFSIN(TFSI2)
      PHI2=CATAN(CFTAN(PSI2)/RIME2)
      XLN2=XLA1+CLLP-CPSIM*(CSIC*(1.-RIME2)-EPS2*XK*(DSIN(2*DSIG)*CCOS(4*
     1FSI2=XLN1+(4*CSIN(CSIG)*CCCS(2*SIG))-CSIC*(2*SIG)-CSIG*(1.+XK/2.)*CPSIM/4.)
      FLAT=FI/2/CETRA
```

138

```
      FLCN=XLN2/CETRA
      GO TC  3ET AZ2:
C
C     INVERSE SCLUTICN.
200   XLN2=FLCN*CETRA
      CXLA=XLA2-XLN1
      IF(CXLA.GT.PI) DXLA=TLFI-DXLN
      CXLN=DSIGN(CMAX1(DAES(CXLN),DBLE(2.E-4)),DXLN)
     +DLB1=CXLA/2.
      PHI2=FLAT*CETRA
      TPSI2=CATAN(PHI2)*RIME2
      TFSI2=CATAN(TFSI2)
10    TEMP=CATAN2((TPSI1-TPSI2)/DTAN(HDLB1),TFSI1+TFSI2)
      XLE2=TEMF-F-+CLEI
      XLB2=TEMF+F+CLEI
      CLB1=CCCS(XLB1)
      FSIA=CATAN(TFSI1/CLE1)
      CPSI2=CCCS((PSIM)
      SSIG1=CATAN2(CPSIM*CSIN(XLB1),CLB1)
      SSIG2=CATAN2(CPSIM*CSIN(XLB2),CCCS(XLB2))
      DSIG=SIG(V+SIGI)/2.
      XK=FLAN*CCCS(2*SIG)*DSIN(DSIG)
      TERM=XK*CCSIM*(CSIC*(2./(RIME2+1.)-XK/2.)-TERM/2.)/2.)/2.
     +DLB2=(CXLA+EPS2*CPSIM*(CSIC*(2./(RIME2+1.)-XK/2.)-TERM/2.)/2.)/2.
      IF(CABS(+DLB2-HDLB1).LT.1.E-7) GC TO 20
     +DLB1=+DLB2
      GC TC 10
20    CKM=BRAVO*(CSIG+TERM-(XK**2)*DSIN(2*DSIG)*CCCS(4*SIG)/8.)/(1.-XK)
      ALFA1=CATAN2(CNE,-CTAN(XLB1)*DSIN(PSI1))
      AZ1=ALFA1
      IF(CKM.LT.C) AZ1=AZ1+FI
      IF(CLEC6).LT.C) AZ1=AZ1/CETFA
      ALFA2=ALFA1
      GC TC 35
35    ALFA2=CATAN2(CNE,DTAN(XLE2)*SPSI2)
     +N2=ALFA2
      IF(CLEC6).LT.C) AZ2=AZ2/CETFA
      CKM=CAES(CKM)
      RETURN
      ENC
C     SUBRCUTINE RAIN1 (N,X,Y,Z,KEYX,KEYY,IER)
CC    THIS RCUTINE PUTS ASCENDING INTEGERS INTO KEYX(N) AND KEYY(N)
CC    SC THAT FAIR3 CAN ARRANGE THE X'S AND Y'S IN ASCENDING CRCER.
C
      CIMENSICN X(N),Y(N),Z(N)
      INTEGER*2 KEYX(N),KEYY(N)
```

RAIN1940
RAIN1950
RAIN1960
RAIN1970
RAIN1980
RAIN1990

139

```
      IER = 0
      IF (N .LE. 32767) GO TO 1
      IER = 1
      RETURN
    1 DO 2 I=1,N
      KEYX(I) = I
    2 Z(I) = X(I)
      CALL RAIN3 (Z,KEYX,N)
      DO 3 I=1,N
      KEYY(I) = I
    3 Z(I) = Y(I)
      CALL RAIN3 (Z,KEYY,N)
      RETURN
      END
C     SUBROUTINE RAIN3 (A,KEY,N)
C     THIS ROUTINE RETURNS THE ARRAYS, KEYX(N) AND KEYY(N), OF INTEGERS
C     WHICH GIVE ASCENDING VALUES OF X(N),Y(N).
      INTEGER*2 KEY(N),IT
      DIMENSION A(N)
    1 M1=N
    2 M=M1/2
      IF (M1 .LE. N) GO TO 1
      M1=M1/2-1
      MAX=((M1/2,1)
      GO TO 2
    2 MM = N/2
      IF (MM .LE. 0) GO TO 6
      K = N-MM
      DO 5 J=1,K
    3 IM=J
    4 IM=IM+MM
      IF (A(IM) .GE. A(IM)) GO TO 5
      TEMP = A(IM)
      IT = KEY(IM)
      A(IM) = A(IM)
      KEY(IM) = KEY(IM)
      A(IM) = TEMP
      KEY(IM) = IT
      IM=IM-MM
      IF (IM .GT.0)   GO TO 4
    5 CONTINUE
      GO TO 2
    6 RETURN
      END
      SUBROUTINE RAIN2 (N,X,Y,Z,KEYX,KEYY,XI,YI,ZI,IER)
      DIMENSION KEY(4),IFC(12),X(12),Y(N),Z(N)
      DIMENSION AMX(3,3),EMX(3),DIST(12),A(9),AX(4)
```

140

```
      EQUIVALENCE (A(1),AMX(1,1)),(NX(1),N1),(NX(2),N2),(NX(3),N3),
     1 (NX(4),N4)
      INTEGER*2 KEYX(N),KEYY(N),NUMOQD(4,3),KEYADS(4),KBOUND(4),
     1 NCIST(12)
      DATA JX/-1/,JY/1/,SIZER/15.0/,JSTCEL/0/
C
C     IER IS THE ERROR RETURN: N3 IS THE MINIMUM NUMBER OF POINTS ALLOW-
C     ED BEFORE INTERPOLATION. N1,N2,N3,N4 COUNT THE NUMBER OF POINTS
C     IN EACH OF THE FOUR QUADRANTS SURROUNDING XI,YI; AND NUMOQD IS A
C     4X3 MATRIX WHICH STORES THE KEY OF EACH POINT USED FOR INTERPOLATION
C     ACCORDING TO THE QUADRANT IT IS IN, AND HOLDS A MAXIMUM OF 3
C     POINTS FOR QUADRANT.
      IER=0
      N3=3
      N1=0
      N2=0
      N3=0
      N4=0
C
      DO 55 J=1,3
      AMX(J) = 0.0
C
      DO 1 I=1,3
    1 AMX(I,J) = 0.0
   55 NUMOQD(I,J)=0
C
C     DEFINE MIN. AND MAX. X AND Y;  DEFINE RANGE X AND RANGE Y;
C     EXTRAPOLATION CONSIDER AREAS WITHIN 5% OF THE BOUNDARIES, AND BOUNDARIES OF
C     EXTRAPOLATION WHICH EXTEND A MAXIMUM OF 5% BEYOND THE RANGE
C     OF POINTS:
      IF (JX.NE.-1) GO TO 8
      JX = 1
      XMIN = X(KEYX(1))
      XMAX = X(KEYX(1))
      YMIN = Y(KEYY(1))
      YMAX = Y(KEYY(1))
      RANGEX = ABS(XMAX - XMIN)
      RANGEY = ABS(YMAX - YMIN)
      AUXX = .05 * RANGEX
      AUXY = .05 * RANGEY
      CNGRXL = XMIN + AUXX
      CNGRXR = XMAX - AUXX
      CNGRYB = YMIN + AUXY
      CNGRYT = YMAX - AUXY
      BNDXL = XMIN - AUXX
      BNDXR = XMAX + AUXX
```

141

```
      ENDYB = YMIN - AUXY
      ENDYT = YMAX + AUXY
C
C  KNUM IS A COUNTER FOR THE NUMBER OF POINTS WE SEARCH THROUGH TO
C  FIND POINTS IN THE CELL.
C  USE A/100 IN PLACE OF A/20 IN THE FOLLOWING STATEMENT WHEN GRID
C  TYPE DATA IS USED.     FOR EXAMPLE:
      KNUM=MAXO((N,A/10))
      KNUM=MAXO((N,A/20))
      KBOUND(1)=0
      KBOUND(2)=C
      KBOUND(3)=-N-1
      KBOUND(4)=-N-1
      NPG=3
      TOL = 1.E-7 * (ABS(XMAX) +ABS(XMIN) +ABS(YMAX) + ABS(YMIN))
C
C  NOW CHECK TO SEE IF XI OF YI ARE WITHIN 5% OF THE RANGE DISTANCE
C  FROM THE BOUNDARY LINES DEFINED BY THE RANGE OF POINTS.
C  NOCFIL IS THE NUMBER OF QUADRANTS WE WANT TO FILL.
      NQDFIL=4
      IF (XI .LE. ENGRXL .OR. XI .GE. DNGRXR) GO TO 11
      IF ((XI .LT. XMIN .OR. XI .GT. XMAX) .OR. (XI .GT. DNGRXR) GO TO 6
      IF ((YI .LT. DNGRYB .OR. YI .GT. DNGRYT) GO TO 10
      IF (YI .LT. YMIN .OR. YI .GT. YMAX) GO TO 7
      GO TO 12
   76
   10 NQDFIL = MAXO (NQDFIL-2,1)
      IER=5
      GO TO 16
   11 NQDFIL = 2
      IER=4
      GO TO 5
C
C  IF THE POINT IS BEYOND OUR RANGE, SEE IF IT IS WITHIN 5% OVER THE
C  BOUNDARY.  IF SO, WE CAN EXTRAPOLATE;  IF NOT THEN ERROR RETURN.
    6 IF (XI .LT. BNDXL .OR. XI .GT. BNDXR) GO TO 66
      IER =7      XI IS BEYOND THE BOUNDARY, BUT 5% OR LESS.
      IER = 15
      GO TO 16
   66 IER=2      XI IS OUT OF RANGE
      RETURN
    7 IF (YI .LT. ENDYB .OR. YI .GT.BNDYT)  GO TO 67
      IER=8      YI IS BEYOND THE BOUNDARY, BUT 5% OR LESS.
      GO TO 16
```

```
C
      IEF =3        YI IS CUT CF RANGE
6     IER = ?
      RETURN
C
C     FIND THE AREA CF OUR X,Y GRID WHERE THE UNKNOWN PCINT XI,YI FITS.
2     IX=KEYX(JX)
      IF(X(IXI.LE.XI) GC TC 3
      JX=JX-1
      IF(JX .GE. 1) GC TC 2
      JX = 1
      GO TC 4
3     IX=KEYX(JX+1)
      IF (X(IX).GE.XI) GO TC 4
      JX=JX+1
      IF(JX .LE. N) GC TC 3
      JX=N
      GO TC 4
4     IY=KEYY(JY)
      IF(Y(IY).LE.YI) GC TC 5
      JY=JY-1
      IF(JY .GE. 1) GC TC 4
      JY = 1
      GO TC 12
5     IY=KEYY(JY+1)
      IF(Y(IY).GE.YI) GC TC 12
      JY=JY+1
      IF(JY .LE. N) GO TC 5
      JY = N
C
C     APC IS THE NUMBER OF PCINTS FOUND IN THE CELL.
12    APC=C
      NQDFIL=MAX(C(NQDFIL-1,1)
      NUMINC=0
C
C     THIS DEFINES THE CELL SIZE.
13    SFA=FLCAT(N)/SIZER
C
C     DEFINE THE CELL LENGTH AND HEIGHT AS A FUNCTION OF THE SMALLER OF
C     THE TWO RANGES.
      IF(RANGEX .GT. RANGEY) XR = RANGEY
      XR = XF/SFA
      YR = XF
C
C     K COUNTS THE NUMBER OF CYCLES OF 4 POINTS WE SEARCH.
      K = 0
```

RAIN3440
RAIN3450
RAIN3460
RAIN3470
RAIN3480
RAIN3490
RAIN3500
RAIN3510
RAIN3520
RAIN3530
RAIN3540
RAIN3550
RAIN3560
RAIN3570
RAIN3580
RAIN3590
RAIN3600
RAIN3610
RAIN3620
RAIN3630
RAIN3640
RAIN3650
RAIN3660
RAIN3670
RAIN3680
RAIN3690
RAIN3700
RAIN3710
RAIN3720
RAIN3730
RAIN3740
RAIN3750
RAIN3760
RAIN3770
RAIN3780
RAIN3790
RAIN3800
RAIN3810
RAIN3820
RAIN3830
RAIN3840
RAIN3850
RAIN3860
RAIN3870
RAIN3880
RAIN3890
RAIN3900

```
C     NOW WE COUNT CYCLES OF 4 POINTS.                              RAIN3510
      KEYADC(1)=KEYX(JX)                                            RAIN3520
      KEYADC(2)=KEYX(JX+1)                                          RAIN3540
      KEYADC(3)=KEYY(JY)                                            RAIN3550
      KEYADS(4)=KEYY(JY+1)                                          RAIN3560
C                                                                   RAIN3570
C     ESTABLISH THE FOUR LINES OF CELL LIMIT. A SQUARE CENTERED ON THE RAIN3580
C     POINT (XI,YI). FOR NOW, XR = YR.                              RAIN3590
      XRT=XI+XR                                                     RAIN4000
      XLFT=XI-XR                                                    RAIN4010
      YTOP=YI+YR                                                    RAIN4020
      YBTM=YI-YR                                                    RAIN4030
C                                                                   RAIN4040
C     THIS LOOP CHECKS TO SEE IF THE 4 POINTS ARE IN THE CELL, IF SO, RAIN4050
C     IT PUTS EACH IN IT'S PROPER QUADRANT.                         RAIN4060
14    DO 21 I=1,4                                                   RAIN4070
C                                                                   RAIN4080
C     CAN'T CHECK THE FIRST CELL SIZE FOR REPETITIONS               RAIN4090
C     IF JX OR JY AREN'T AT THE BOUNDARIES OF THEIR ARRAYS,         RAIN4110
C     OR IF THE CELL SIZE HASN'T BEEN INCREASED.                    RAIN4120
C     OR IF THERE ARE NO POINTS IN THE CELL...                      RAIN4130
C                                                                   RAIN4140
C     KEYADS IS THE KEY ADDRESS, KBOUND IS THE BOUNDARY OF X AND Y  RAIN4150
C     ARRAYS, NUMINC IS THE NUMBER OF CELL SIZE INCREASES.          RAIN4160
      IF (KEYADS(I).LE.KBOUND(1)) GO TO 21                          RAIN4170
      IF (NUMINC.EQ.0.AND.NPC.EQ.0) GO TO 16                        RAIN4180
C                                                                   RAIN4190
      DO 15 J=1,NPC                                                 RAIN4200
      IF (IFC(J).EQ.KEY(I)) GO TO 21                                RAIN4210
15    CONTINUE                                                      RAIN4220
C                                                                   RAIN4230
C     SEE IF THE POINTS ARE IN THE CELL.                            RAIN4240
16    XVALUE=X(KEY(I))                                              RAIN4250
      YVALUE=Y(KEY(I))                                              RAIN4260
      IF (XVALUE.GT.XRT.OR.XVALUE.LT.XLFT.OR.YVALUE.GT.YTOP.OR.YVALUE.LT RAIN4270
     1.YBTM) GO TO 21                                               RAIN4280
C                                                                   RAIN4290
C     IF YOU GET THIS FAR, THE POINT IS IN THE CELL,                RAIN4300
C     YOU CAN TAKE A COFFEE BREAK NOW.                              RAIN4310
C                                                                   RAIN4320
C     PLACE THE POINT IN IT'S PROPER QUADRANT USING A MAXIMUM OF NPQ RAIN4330
C     POINTS PER QUADRANT. FOR NOW, NPQ = 3.                        RAIN4280
```

144

```
C     ALPCD(I,J) HCLCS THE KEY CF A POINT IN QUADRANT I, AND MAXIMUM J
C     IS (3 FCINTS PER QUACRANT).
      IF (XVALLE.LE.XI) GC TC 18
      IF (YVALLE.LE.YI) GC TC 17
      IF (N1.GE.3) GO TO 21
      NUMOCC(1,N1+1)=KEY(I)
      N1=N1+1
      GO TC 20
17    IF (N4.GE.3) GO TO 21
      NUMOCC(4,N4+1)=KEY(I)
      N4=N4+1
      GO TC 20
18    IF (YVALLE.LE.YI) GC TC 19
      IF (N2.GE.3) GO TO 21
      NUMOCC(2,N2+1)=KEY(I)
      N2=N2+1
      GO TC 20
19    IF (N3.GE.3) GO TO 21
      NUMOCC(3,N3+1)=KEY(I)
      N3=N3+1
20    NPC=NPC+1
21    IFC(NPC)=KEY(I)
      CONTINLE
C
C     IF NQDFIL CLADRANTS ARE FILLED, INTERPCLATE.
C     ACWPTS IS THE NUMBER CF CUADRANTS WITH PCINTS IN THEM.
      NQWPTS=C
      DO 22 I=1,4
      IF (NUMCCC(I,1).NE.C) NQWPTS=NQWPTS+1
22    CCNTINLE
C
      IF (NCWFTS.GE.NQDFIL.AND.NPC.GE.NN3) GO TO 25
      K = K+1
C
C     IF K.GE. KNUM WE'LL INCREASE CELL SIZE. OTHERWISE,
C     GET THE NEXT FCUR FCINTS, GC BACK TC SEE IF ANY OF THESE ARE IN
C     THE CELL.
      IF (K.GE.KNUM) GO TC 28
      KEYACS(1)=PA>C(JX-K,C)
      IF (KEYACS(1).EQ.0) GC TC 24
      KEYACS(1)=KEYX(KEYACS(1))
24    KEYACS(2)=MINC(JX+14K,N+1)
      IF (KEYACS(2).EC.N+1) GO TO 25
      KEYACS(2)=KEYX(KEYADS(2))
25    KEYADS(2)=-KEYADS(2)
```

145

```
      KEYADS(3)=MAXC(JX-1,C)                                  RAIN4870
      IF(KEYADS(3).EQ.0) GC TC 26                             RAIN4880
      KEYADS(3)=KEYY(KEYADS(3))                               RAIN4890
26    KEYADS(4)=MINC(JX+1+K,N+1)                              RAIN4900
      IF(KEYADS(4).EQ.N+1) GO TO 27                           RAIN4910
      KEYADS(4)=KEYY(KEYADS(4))                               RAIN4920
27    KEYADS(4)=-KEYADS(4)                                    RAIN4930
      GO TO 14                                                RAIN4940
C                                                             RAIN4950
C     THIS INCREASES OUR CELL SIZE.                           RAIN4960
C     JSTCEL COUNTS THE NUMBER OF POINTS INTERPOLATED WITHOUT A CELL   RAIN4970
C     INCREASE. WHEN IT REACHES 3, THE CELL SIZE IS DECREASED.         RAIN4980
28    SIZER=1.1*SIZER                                         RAIN4990
      JSTCEL=C                                                RAIN5000
      NUMINC=NUMINC+1                                         RAIN5010
C                                                             RAIN5020
C     SETTLE FOR ONE FEWER QUADRANTS AT THIS POINT,           RAIN5030
C     OR GO BACK FOR MORE POINTS.                             RAIN5050
      IF (NQDFTS.LT.NCDFM1.CF.NPC.LT.NN3) GO TO 12            RAIN5060
C                                                             RAIN5070
C     NOW FIT THE LEAST SQUARES PLANE THROUGH THE POINTS TO CALCULATE  RAIN5080
C     A VALUE FOR Z POINTS IN ALL 4 QUADRANTS GO TC 27. CTHERWISE TAKE RAIN5090
C     IF THERE ARE 2 POINTS, PUT A LEAST SQUARES PLANE THROUGH THEM    RAIN5100
C     AND EXTRAPOLATE.                                        RAIN5110
25    IF (NQDFTS.EQ.4) GO TC 27                               RAIN5120
      DO 30 I=1,NPC                                           RAIN5130
      K2=IFC(I)                                               RAIN5140
30    DIST(I)=SGRT((XI-X(K2))**2+(YI-Y(K2))**2)              RAIN5160
      DO 31 I=1,APC                                           RAIN5170
31    NDIST(I)=I                                              RAIN5180
C                                                             RAIN5190
      CALL RAIN3 (DIST,NCIST,NFC)                             RAIN5200
C                                                             RAIN5210
C     PUT THE 3 CLOSEST POINTS INTO A MATRIX.                 RAIN5220
C                                                             RAIN5240
C     LOOPING FROM 3 TO NPC WILL CONTINUE UNTIL A NON-COLINEAR SET OF  RAIN5250
C     POINTS IS FOUND. IF ALL POINTS ARE COLINEAR, MORE POINTS MUST BE RAIN5260
C     SEARCHED.                                               RAIN5270
      IHOPE = 3                                               RAIN5280
C                                                             RAIN5290
55    K2=IFC(NDIST(1))                                        RAIN5310
      AMX(1,1) = X(K2)                                        RAIN5320
      AMX(1,2) = Y(K2)                                        RAIN5330
```

146

```fortran
      AMX(1,1) = 1.0
      BMX(1) = -Z(K2)
      K2=IFC(ACIST(2))
      AMX(2,1)= X(K2)
      AMX(2,2)= Y(K2)
      AMX(2,3)= 1.0
      BMX(2)= -Z(K2)
      K2=IFC(ACIST(IHGPE))
      AMX(3,1)= X(K2)
      AMX(3,2)= Y(K2)
      AMX(3,3)= 1.0
      BMX(3)= -Z(K2)
      GO TO 55
C
C     IF ALL ELSE FAILS, COME HERE. THE 3 POINTS ARE CO-LINEAR.
C     IF THE POINTS ARE COLINEAR, ANOTHER POINT MUST BE CHOSEN, OR
C     MORE POINTS MUST BE SEARCHED FOR NON-COLLINEARITY.
   32 IER=6
      IF (IHCFE .LT. NPC)  GO TO 33
      ANS=AA*41
      GO TO
   33 IHCFE = IHCFE + 1
      IF (IHCFE .EC. 7)  GO TO 68
      GO TO 55
C
C     IF WE GET HERE, TOO MANY POINTS ARE COLINEAR,   ERROR RETURN.
   68 IER=5
      RETURN
C
C     IF THERE ARE POINTS IN ALL 4 QUADRANTS, TAKE THE NEAREST ONE IN
C     EACH QUAD. AND FIT A LEAST SQUARES PLANE THROUGH THEM.
   37 DO 44 I=1,4
      K2=ALPCGC(I,1)
      NQ=ALPCGC(I,J)
      IF (NG .EC. 1) GO TO 39
      TQ = SCRT ((XI-X(K2))**2 + (YI-Y(K2))**2)
C
C     SEARCH FOR THE CLOSEST POINT IN QUADRANT I.
   40 K3 = K2+AC
      K3 = ALPCGD(I,J)
      TPC = SCRT ((XI-X(K3))**2 + (YI-Y(K3))**2)
      IF (TPC .GE. TQ)  GO TO 41
      TQ = TPC
      K2 = K3
   41 CONTINUE
```

147

```
 35    AMX(1,1) = AMX(1,1)  + X(K2)**2
       AMX(2,1) = AMX(2,1)  + X(K2)*Y(K2)
       AMX(3,1) = AMX(3,1)  + X(K2)
       AMX(2,2) = AMX(2,2)  + Y(K2)**2
       BMX(1)   = BMX(1)    + X(K2)*Z(K2)
       BMX(2)   = BMX(2)    + Y(K2)*Z(K2)
       BMX(3)   = BMX(3)    + Z(K2)
 44    CONTINUE
       AMX(1,2) = AMX(2,1)
       AMX(1,3) = AMX(3,1)
       AMX(3,2) = AMX(3,2)
       AMX(3,3) = 4.

C      SOLVE THE 3 EQUATIONS & FIND THE PLANE PASSING THROUGH THEM,
C      THEN EXTRAPOLATE.
C
C      THE FOLLOWING FUNDREC OR SC STATEMENTS SOLVE THE SET OF SIMUL-
C      TANEOUS EQUATIONS, AMX * X = BMX.
C
C      FORWARD SOLUTION
C
 58    JJ=-3
       DO 65 J=1,3
       JJ=J+JJ+1
       JJ=JJ+4
       BIGA=0.
       IT=J-1
       DO 70 I=J,3

C      SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN

       IJ=IT+I
       IF(ABS(BIGA) .GE. ABS(A(IJ))) GO TO 70
       BIGA=A(IJ)
       IMAX=I
 70    CONTINUE

C      TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)

       IF (ABS(BIGA) .GT. TOL) GO TO 80
       GO TO 132

C      INTERCHANGE ROWS IF NECESSARY

 80    I1=J+3*(J-2)
       IT=IMAX-J
       DO 85 KK=J,3
```

```
      I1=I1+3
      I2=I1+IT
      SAVE=A(I1)
      A(I1)=A(I2)
      A(I2)=SAVE
C
C     DIVIDE EQUATION BY LEADING COEFFICIENT
C
   90 A(I1)=A(I1)/BIGA
      SAVE=BMX(IMAX)
      BMX(IMAX)=BMX(J)
      BMX(J)=SAVE/BIGA
C
C     ELIMINATE NEXT VARIABLE
C
      IF (J .EQ. 3) GO TO 110
      IQS=3*(J-1)
      IQX=IQS+IX
      IT=J-IQX
      DO 100 JX=X+IT,JYY,3
      IXJX=IXJX-1)+IX
      JJX=IXJX+3*(JX-1)+IX
      A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
  100 BMX(IX)=BMX(IX)-(BMX(J)*A(IXJ))
  ...
C
C     BACK SOLUTION
C
  110 DO 120  J=1,2
      IA=4-J
      IB=3-J
      IC=3
      DO 120  KK=1,J
      BMX(IB)=BMX(IB)-A(IA)*BMX(IC)
      IA=IA-1
  120 IC=IC-1
C
C     THIS IS THE ANSWER.
C     NOW WE COMPLETE THE CALCULATION OF ZI.
C
      ZI = -BMX(1)*XI -BMX(2)*YI -BMX(3)
C
C     IF 3 POINTS ARE CALCULATED WITHOUT A CELL INCREASE, DECREASE IT.
C
      JSTCEL=JSTCEL+1
      IF (JSTCEL.LT.3) RETURN
      SIZER=SIZER*0.9
      JSTCEL=0
      RETURN
C     DEBUG INIT, SUBCHK, TRACE
```

FAIN677C

```
      AT 2
      TRACE ON
      AT 120
      TRACE OFF
      END
C/*
//GO.FTC1FCC1 DD  UNIT=2330,VOL=SER=DISK03,DSN=S2648.CONF30,
//   DISP=SHR
//GC.FT01F0G2 DD  UNIT=2330,VOL=SER=DISK03,DSN=S2648.CONF24,
//   DISP=SHR
//GG.FT02F0G1 DD  UNIT=3330,VOL=SER=DISK01,DISP=SHR,DSN=S2648.GRID2
//GC.SYSIN  DD  *
EAST LOFOTEN BASIN TO VESTERAALEN RISE
CENTER LAT/LON 70-04,15-51
70.C7C5,15.85C22  1.C  75.
    5    2
C/*
```

SYNGEN

```
************************************************************

************************************************************

C//SYNGEN JOE (    ,0354,1X82),'ERAZO',TIME=2
C
C     USER MAY GENERATE A SYNTHETIC SEA BOTTOM. THE FOLLOWING TYPES
C     TYPES ARE AVAILABLE: PARABOLIC BASIN, CONICAL SEAMOUNT, WEDGE,
C     TROUGH, RIDGE, AND UNDULATING BOTTOM.
C
//FORT.SYSIN DC *
//EXEC FORTCLG,REGION.GC=250K
      REAL*8 TTL(12),DHN,SYN(17)
      DATA SYN/'         ','SYNTHETI','C SEA BE','D
     1*       ','COMPUTER',' GENERAT','ED     ',
     2*RIDGE  ','TROUGH  ','SEAMOUNT','UNDULAT.','BASIN  '/
      DIMENSION ZE(151,151)
      READ(5,5000)CNLAT,CNLCN,DMIN,DMAX,ITYPE
5000  FORMAT(4F10.5,I5)
      CDIF=CDMAX-DMIN
      DO 10 I=1,11
      TTL(I1)=SYN(I)
10    TTL(1(2)=SYN(ITYPE)
      IF(ITYPE.NE.13) GO TO 13
      IMAX=11
      JMAX=11
      CHX=1E.
      DO 50 J=1,11
50    ZB(I)=DMAX
      DO 130 J=1,11
130   ZB(6,J)=DMIN
      GO TO ...
C
C     WEDGE: TYPE 12
12    IF(ITYPE.NE.12) GO TO 15
```

151

```
      IMAX=11
      JMAX=11
      SLOPE=DIF/S.
      DO 120 I=1,110
      ZB(I,1)=ZMAX-SLOPE*(I-1)
      ZB(I,110)=ZMIN
  120 CONTINUE
   20 GO TO ...
C
   15 IF(ITYPE.NE.15) GO TO 16
      IMAX=151
      JMAX=151
      SLOPE=DIF/75.
  151 DO 150 I=1,151
      YSG=(J-76)*1.1E1**2
      XSG=(I-1)*1.1E1**2
      FSQ=XSG+YSG
      ZB(I,J)=ZMIN+SLOPE*SQRT(FSQ)
      GO TO ...
C
   16 IF(ITYPE.NE.16) GO TO 14
      IMAX=151
      JMAX=151
      SLOPE=DIF/2.
      I=1,151
      DEP=ZMIN+SLOPE*(COS((I-76)*0.2094395)+1.)
      DO 160 J=1,151
      ZB(I,J)=DEP
      GO TO ...
C
   14 IF(ITYPE.NE.14) GO TO 17
      IMAX=11
      JMAX=11
      CHN=10  J=1,11
      DO 40 I=1,11
      ZB(I,40)=ZMIN
   40 ZB(I,40)=DEP
  140 GO TO ...
C
   17 IF(ITYPE.NE.17) GO TO 151
      IMAX=151
      JMAX=151
```

```
      CTN=1.
      SLCFE=CIF/5625
      COS1C=1.+1.*IE1
      YSC=1.-7E1.*IE1
      COS1C=1./1.*IE2
      XSC==XSC1+YSC
      RSC==1.+(I-I)*C
170   ZB(I,J)=IMAX-SLCPE*FSC
SSS   WRITE(ICCC)TTL,CNLAT,CNLCN,DHN,IMAX,JMAX
2000  FORMAT(TE,6A8/6A8/3F10.;,2IE)
      WRITE(ICCO) ((ZB(I,J),I=1,IMAX),J=1,JMAX)
100C  FORMAT(ICFE.4)
      ENCFILE
6000  PRINT(CCCC,TTL,CNLAT,CNLCN,IMAX,JMAX,DMIN,DMAX
      FORMAT(10X,'A SYNTHETIC SEA BED TITLED',:2(T40,6A8/)/10X,
     1        'CRS TRUCTEC AFCURC LAT=',F10.5,' LCN=',F10.5/10X,
     2        'GRIC CIMENSIONS ARE',I3,' BY',I3,'.',I3,', DEPTHS RANGE FRCM',F6.2,
     3        ' TC',F6.3,' KM.')
      STCP
      ENC

C/*
//GC.FT02FCC01 DC UNIT=333C,VCL=SER=DISKO3,DSN=S2648.SYNBED16,
//  LCB=(RECFM=FE,LRECL=8C,BLKSIZE=800),DISP=SHR,
//  SFACE=(TRK,(20,5))
//GC.SYSIN CC *
85.3        C.           1.           5.          12
C/*
```

153

APPENDIX K

```
**********************************************************
```

G3DP

```
**********************************************************
```

SUBROUTINE GRDSCT IS CONTAINED IN THIS LISTING
IT IS CALLED BY GRECK AND SUBROUTINE TRACER

```
C//G3DP2   JCB   (      ,20354,1X82),'ERAZO  SMC1704',TIME=2
// EXEC   FORTCLGH,REGION.GO=320K
//FORT.SYSIN  DC  *
      REAL*8  TTL(12),CNX,CNY,CNLAT,CNLNG,DHN
      COMMON /CIN2/ KMX,KMY
      COMMON /FLGTIT/ TTL,CNX,CNY,CNLAT,CNLNG
      LOGICAL*1 ICA(76,46),LHAF
      DIMENSION  ZE(151,51),CC(76,46),F(2),SIZE(2),
     1  WK(76,46),KX(2000),KY(2000),X(76),Y(46)
      AECVE  DIMENSIONS SHOULD EXACTLY EQUAL THE FOLLOWING:
     1.  ZE(NRCW,KMY)
     2.  CC(NRCW,NCCL), WK(NROW,NCCL,3), X(NROW), Y(NCCL),
        ICA(NRCW,NCCL) SUCH THAT
        NROW=INT((KMX+1)/2), NCOL=INT((KMY+1)/2)
        AFCW=INT((KMX+1)/2)
C       BELOW DECLARATIONS MUST CONFORM.
      DATA F/2*0./
      KMX=151
      KMY=51
      ARCW=(KMX+1)/2
      ACCL=(KMY+1)/2
5300  FCRMAT(5,5CCC)( XST,YST,ZENC,DEGUP,DEGCH,LH
      LH=MAXC(LH,-6)
      LW=MINC(LH,18)
      SIZE(1)=LW
      SIZE(2)=LH
      CALL  GRDSCT(XST,YST,XMAX,YMAX,IMAX,JMAX,DHN,ZB)
      ICT=2*ARCCL+1
      JCT=2*NCCL+1
```

154

```
30 DD 10 I=1, NRCW
   II=ICT-2*I    ACCL
   DD 20 J=1, JCCL
   JJ=JCT-2*J
20 D(II,JJ)=(ZEAC-ZB(II,JJ))*3
10 X(II)=I
   Y(II)=I
   CALL FLT3D1(X,NROW,Y,NCOL,D,DFGUP,DEGCCW,F,TTL,SIZE,WK,ICN,
  1 WKX,WY,2CCO,2)
   STOP
   END
C/*
//LINK.USCC DD UNIT=3330,VCL=SER=DISK01,DSN=S2648.ECTLWCD,
// DISF=SHF,LAEEL=(,,,IN)
//LINK.SYSIA DD *
   INCLUDE USCC(GRCSCT)
   ENTRY MAIA
C/*
//GC.FTO1FOO1 DD DSN=S2648.GRID2,UNIT=3330,VCL=SER=DISK01,
// DISP=SHF,LAEEL=(,,,IN)
//GC.SYSIN DD *
   0.    4.    3C.    45.    6
C/*
```

```
C ****************************************************************
C *
C *
      SUBROUTINE: GRDSCT
C *
C ****************************************************************
C *
      SUBROUTINE GRDSCT(XST,YST,XMAX,YMAX,IMAX,JMAX,DHN,ZB)
C *
C     EXTRACTS KMX*KMY SECTION CF LARGER 2C DATA MATRIX.
C     XST,YST MUST BE MULTIPLES CF 10*DHN.
C     IMAX,JMAX MUST PE CF XST+KMX, YST+KMY, RESPECTIVELY.
C
C     GRDSCT IS CALLED BY THE FOLLOWING PROGRAMS:
C                                 GRDCP
C                                 GRDCTK
C                                 ECTRACE -CALLED BY TRACE
C
      REAL*8 TTL(12),CNX,CNY,CNLAT,CNLNG,DHN
      COMMON /FLOTIT/ TTL,CNX,CNY,CNLAT,CNLNG
      COMMON /DIM/ KMX,KMY
      DIMENSION ZB(KMX,KMY), C10(10)
      READ(1,1C10) TTL,CNLAT,CNLNG,DHN,IMAX,JMAX
 1010 FORMAT(6A8,/6A8,/3F10.5,2I5)
      CNX=(IMAX-1)*DHN/2.
      CNY=(JMAX-1)*DHN/2.
      IF(KMX.EC.IMAX.AND.KMY.EC.JMAX) GO TO 80
C     READ ONLY A SECTION CF INPUT GRID.
      FAR=ELT
      KMX=PIAC(KMX,IMAX)
      KMY=PIAC(KMY,JMAX)
      XST=AMIN1(XST,(IMAX-KMX)*FAR)
      YST=AMIN1(YST,(JMAX-KMY)*FAR)
      GRFL=DHN
      XST=GRFL*IFIX(XST/GRFL)
      YST=GRFL*IFIX(YST/GRFL)
C     NC = CF CARDS REPRESENTING CCLUMNS (Y-VALUES) SKIPPED
      NC = CF ITEMS SKIPPED WEST (LEFT)
      KXSACLMY) ITEM THRESHCLC, DUMMY CONTAINS NO NEEDED DATA (SKIP
      KSACLMY) ITEM THRESHCLC, DUMMY FULL CF NEEDED DATA (SKIP NO
      CLMMY) ITEM THRESHCLC, DUMMY FULL OF NEEDED DATA (SKIP NO
```

```
      NCSS=IMAX*IFIX(YST/ERFL)
      NISW=IFIX(XST)
      KKSAD=KMX*X
      KKSAD=KKAC+KMX
      J=1
      ITER=0
 4    IF(NC)5,10,10,4
 10   NISW=IFIX=12NCSS
      IZB=0
      IL=1
 1000 READ(1,1000,ERR=70,END=70)
 1000 FORMAT(IC*F6.4)
      ITER=ITER+10
 15   IF(ITER.LE.NISH) GO TO 20
      IF(ITER.LE.NISAD) GO TO 50
      IST=IST+1
      ID=IC*IL-IST+1
      IF(ITER.GT.KSND) IC=1
      IB=IB+1
      IHC(I)=:CIO(IC)
      IHC(I)=EC.KMX) GO TO 60
 30   IZE=IST
      IL=IL+1
      IF(ITER.LT.IMAX) GO TO 20
 40   ITER=ITER-IMAX
 50   IZB=0
      GO TO 15
      J=J+1
      IF(J.GT.JMAX) GO TO 70
      IF(J.LE.KMX) GO TO 4C
 60   READ(1,1000) Z8
      REAL ENTIRE INPUT CRIC
      J=KMX+1
 70   J=J-1
      ITER=(KMX-1)+CLA
      XRAN=(KMX-1)*XRAN
      XMAX=XST+XRAN
      YMAX=YST+YRAN
      PRINT(6,CCRIC,TTL,CNLAT,CNLNG,CNX,CNY,XST,XMAX,YST,YMAX,ITEMS
 6000 FORMAT(*CCRIC FILE CALLEC ',52(*/T19,'*'),2(/T19,'*',1X,6A8,1X,'*')/T19,
     1'* HAS BEEN REAC. CENTER LAT/LONG IS ',F10.5,'/',F10.5,
     2'* AT ',F7.3,'/',F7.3,' KM.'/' RANGES ARE ',F6.1,'TO ',F6.1,
     3' KM IN X ANC ',F6.1,'TO ',F6.1,' KM IN Y (',I5,' ITEMS).')
```

CFC01000
CFC01010

RETURN
END

APPENDIX L

****

GRDCHK

SUBROUTINES:
CORNER
GECPLR
GEODST

****

```
C //GRDCHK2 JOB  ( ,0354,1X82),'ERAZO  SMC 1704',TIME=2
// EXEC FORTCLGW,REGION.GO=350K
//FORT.SYSIN DD *
      IMPLICIT REAL*8(B-H),REAL*8(C-W)
      LOGICAL*1 LTG(3),LFLOT,LGEO
C SET DIMENSIONS: ZB(KMX*KMY),ZV(KMX*KMY+4) AS KMX,KMY DECLARED BELOW.
      DIMENSION ZV(22805),ZB(151,151),AX(183),AY(183),TTL(12)
      COMMON /CIRC/ TTL,CAX,CAY,CNLAT,CNLON
      COMMON /FLOT/IT/ TTL,CAX,CAY,XMIN,XMAX,YMIN,XMAX,YMAX,XRG,YRG,APDV
      DATA LASK/Z1111/
      CINT(CFAFG) = AINT(SNGL(CPARG))
      KMX=151
      KMY=151
C CHECK ABOVE PARAMETERS WITH DIMENSION REQUIREMENTS.
      READ(5,5000) NL,LEN
      READ(5,5001) ZST,ZEND,XMIN,YMIN
      LFLOT=.FALSE.
      READ(5,5000,END=20) IPS,IPE,JPS,JPE
      LGEO=.TRUE.
5000  FORMAT(16I5)
5001  FORMAT(4F6.1)
20    IANL=IABS(NL)
```

```
2DV=(ZFAC-ZST)/IANL
CALL GRCC(CT(XPIN,YPIN,XMAX,YMAX,IMAX,JMAX,C+N,ZB)
XRCWL=IAC(IFF-IDS+1,KMX)
JPEC=IRG(IFF-JPS+1,KMY)
XRG=XPXPZIAC
LGEC=IAE(LFAC)CT=0
LEA=IAE=LFEN=2)
LPLCT=LEN.GE.4.AND.IANL.LE.1CO
LEA=CLL FLCWS(0.,0,0)
LEA=XLER.19)
AFLCT=LFAC-CT=0
YOR=(2I-LEN)/2.
CALL FLCWS(1F.,YOR,-2)
AXY=AXAXI(15,,AMAXI(XRG,YRG))
APLCV=A4I7(ARI5)
AFACT=AIAFCCV/ARI5
AFACT=AFACTCR(AFCT)
IFACT.LCT.LE.C)GO TC 25
XPEI=IAC+LEN+(IPS-1)
XPEI=IAC+LLN+(IPE-1)
PRINT 6C10,IPS,IPE,XFS,XFE,C+N
FORMAT(CF FEACT BY COLUMNS; RCWS ',I3,' TC ',I3,'; DHN=',F5.1,' TO ',F5.1,'; KN,',I3,'//)
AICV=AIAFACT(AFCW-1),1.5.)
AICV=AMAXI(AICV,1.)
AXCV=AAIICV
FVAL=IFC,
XMCVE=C,
AX=0
CALL AXIS(0.,0.,'I-VALUES',8,15,,90,,FVAL,AICV)
CALL FLCT(C,,15,,2)
CALL FLCT(0.,0,,-2)
IRS=IFS,TRE
CC7C=IF=IIFS,TRE
ZV(II+1)=0AICV
ZV(II+2)=AICV/E,
AZCV=ZRG+AFCH-1
IVS=IVE+1)=0,0
ZV(IVE+1)=0,0
ZV(IVE+2)=AZCV
```

160

```
      DO 1C JJ=1,NCCL
      J=JJ+1
      YKR=YIN+DHN*(J-1)
6020  PRINT(*C2C,J,YKR
      FCRMAT(' CCLUMN ',I3,' Y=',F5.1)
6030  PRINT(*C3C,(ZB(I,J),I=IPS,IPE)
      FCRMAT(1X,22F6.3)
      IF(IVF-1)...,I=IVS,IVE
6C    ...
      CALL LINE(ZV(IVS),ZV(IPS),NROW,1,0,0)
      CALL WHERE(XNOW,YRCW,AFCT)
      CALL SYMBCL(XNCW,YRCW,.08,'J=',90.,2)
      CALL NUMBER(999.,999.,.08,FPN,90.,-1)
      CALL SYMBCL(999.,999.,.08,',Y=',90.,4)
      CALL NUMBER(999.,999.,.08,YKM,90.,0)
      CXZXX=XACVE+.2
      CALL FLCT(.2,0.,-3)
10    XMCVE=XRCVE+5.
      XY=IFIX(XRG/APDV+.3)
      AX=NX+XMCVE-5
      CALL AXIS(0.,0.,'X-VALUES',-8,15.,90.,XPS,AXCV)
      CALL FLCT(C.,15.,2)
      CALL GRIC(-XMCVE,0.,-NX,.2,NY,1.,LMASK)
      IPEN=999
      IF(FLCT)IPEA=-IPEN
      CALL FLCT(C.,C.,IPEN)
      IF(.NCT.FLCT)STCF
C     TRANSFOSE ARRAY TC PUT NCRTH AT TCF
25    DO 4C K=1,KMY
      I=I-KM
      DO 4C J=1,KMX
40    ZV(IJ)=ZB(I+KM*(J-1)+I
      LTG(I)=.TRUE.
      LTG(II)=.FACT.LCEG
      IF(LTG(II))CC TC 5C
      XCR=LEN+IVF
C     FLCT GECCRAPHIC BCCACARY LINES.
      CALL FLCTS(C,0,3)
      CALL FLCT(XCF,YCR,-2)
      CALL SYMBCL(.3,-.£,.CE,TTL,90.,48)
```

```
      CALL SYMBOL(.4,-.5,.,CE,TTL(7),90.,48)
      CALL FACTCR(AFCT)
      CALL CCRREF(GLATMN,GLATMX,GLCNMN,GLONMX,IMAX,JMAX)
      CLTC=1.25
      ABL=CMAX1(CABS(GLATMX),CABS(GLATMN))
      CLN=2C.
      IF(ABL.LT.85.) DLN=CLN/2
      IF(ABL.EC.) DLN=CLN/2
      CLNC=1.
      IF(ABL.GT.75.) GO TC 210
      CLNC=1.
  21C CLNC=.25
      SLCN=GLCNMN-CLN
      ELCN=GLCNMX+DLN
      SLAT=GLATMN-CLT
      ELAT=GLATMX+DLT
      IF(SLCN.LT.CNLCN-SC.) SLCN=SLON+DLN
      IF(ELCN.GT.CNLON+90.) ELCN=ELCN-DLN
      SLAT=CMAX1(SLAT,DBLE(-85.))
      ELAT=CMIN1(ELAT,CBLE(+85.))
C     DRAW LATITUDE LINE
      CJ=CLTC
      GI=DINT(SLAT)
      GJ1=DINT(SLCN)
      IF(DLN.NE.1.) GJ1=1C*DINT(GJ1/10.)
      GJ2=ELCN
  34C CALL CECFLR(GI,GJ1,GJ2,C.,AX,AY,1)
      GI=GI+CLT
      IF(GI.LE.ELAT) GC TC 240
C     DRAW LCNGITUDE LINE
      GI=CLTC
      GI=DINT(SLAT)
      GJ1=DINT(SLAT)
      GJ2=ELAT
      IF(GJ1.CE.GJ2) GC TC 50
  350 CALL CECFLR(GI,GJ1,GJ2,D.,AX,AY,2)
      GI=GI+CLN
      IF(GI.LE.ELCN) GC TC 350
   5C AX(1)=25T
      IF(NL.LT.2) GC TO 35
      CO 30 I=2,NL
      AX(I)=AX(IM1)+2CV
   30 ...
   35 IW=IFIX(XRG*AFCT/AFCV+.1)
      IH=IFIX(XRG*AFCT/AFCV+.1)
      CALL CCNTLR(ZV,KMY,KNX,KMY,AX,NL,TTL,LEN,LFN,LTG)
      STCP
```

```fortran
      ENC

      SUBROUTINE CORNER(GLATMN,GLATMX,GLONMN,GLONMX,IMAX,JMAX)

C     CALCULATES GRID CORNER COORDINATES.
      IMPLICIT REAL*8(A-H),REAL*8(C-W)
      DIMENSION TTL(12)
      COMMON /FLCTIT/ TTL,CAX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YPG,AFCV
      DIST=DC*(CNX**2+CNY**2)
C     CALCULATE LAT/LON OF GRID CORNERS. C1 IS SW AND C2 IS SE.
      HS=DATAN2(CNX,CNY)
      CALL GEODST(CNLAT,CNLON,C3Y,C3X,HS,HF,DIST,1)
      HS=DATAN2(CAX,-CNY)
      CALL GEODST(CNLAT,CNLON,C2Y,C2X,HS,HF,CIST,1)
      C1Y=C2Y
      C4Y=C3Y
      C4X=CNLCN-CNLCN
      C1X=C3X-CNLCN
      C1X=C2X-CNLCN
C     CALCULATE BOUNDARY LATS/LONS THAT FORM SPHERICAL RECTANGLE
C     WHICH INCLUDES GRID.
      IF(CNLAT.GE.0.) GO TC 5
      GLATMX=C3Y
      GLATMN=C1Y
      GLONMX=C2X
      CALL GEODST(CNLAT,CNLON,180.,HFC,CNY,2)
      GO TC 6
    5 GLATMX=C2Y
      GLATMN=C4Y
      GLONMX=C3X
      CALL GEODST(CNLAT,CNLON,CLATMX,CNLON,0.,HF,CNY,1)
    6 PRINT 6010,CNLAT,CNLON,CLATMX,CNLON,YMIN,C1Y,C1X,XMAX,YMAX,
     1CJY,C2Y,C4Y,C4X,GLONMN,GLCNMX,C2X,CLATMN,GLATMX,IMAX,JMAX
 6010 FORMAT(/,TI1,'VALUES AT GRID CORNERS:'/T21,X,'T31,'Y',T40,
     1'LAT',T50,'LON'/4(17X,4F10.5)/T14,'MIN LCN',3X,'MAX LCN',
     23X,'MIN LAT',3X,'MAX LAT',3X,'IMAX',3X,'JMAX'/TI1,4F10.5,2I7/)
      RETURN
      ENC

      SUBROUTINE GEOPLP(GI,CJ1,CJ2,DJ,XNEW,YNEW,LLL)

C     IMPLICIT REAL*8(C-H),REAL*8(C-W)
```

```
      DIMENSION TTL(12),XNEW(183),YNEW(183)
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APOV
      INTEG=LLL+2
      IF(LLL.NE.1) GO TO 10
C          PLOT LATITUDE LINE GI FROM LONGITUDE GJ1 TO GJ2
      PLAT=GI
      PLON=GJ1
      GLON=GJ2
      GLON=0.
      GO TO 20
   10 PLAT=GJ1
C          PLOT LONGITUDE LINE GI FROM LATITUDE GJ1 TO GJ2
      PLON=GI
      GLON=GJ2
      GLAT=0.
      NPTS=0
   20 CALL GEODST(CNLAT,CNLON,FLAT,PLON,HS,HF,DIST,-1)
      IF(FLAT.GT.CLAT.OR.FLON.GT.GLON) GO TO 20
      NPTS=NFT+1
C          MUST BE TRANSPOSED TO LIE ON ROTATED CONTOUR PLOT.
      YNEW(NFTS)=DIST*DSIN(HS)+CNX
      XNEW(NFTS)=YRG-(DIST*DCOS(HS)+CNY)
      FLAT=FLAT+CLAT
      FLON=FLON+CLON
      GO TO 20
   30 APFF1=NFTS+1
      APF2=NFTS+2
      YNEW(NFFI))=XMIN
      YNEW(NFF2))=APOV
      XNEW(NFFI))=YRG-YMIN
      XNEW(NFF2))=APOV
      CALL LINE(XNEW,YNEW,NFTS,1,1,INTEO)
      RETURN
      END

      SUBROUTINE GEODST(SLAT,SLON,FLAT,FLON,AZ1,AZ2,DKM,MODE)
C
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      LOGICAL LFEG
      DATA EPS/.C81819356/,ERAVO/6356.75023/
      FUNK(SPSI)=DTAN(DATAN(EPS*SPSI/R1ME21/2.)**2
```

164

```
      CNE=1.+CATAN(CNE)*2
      HAPI=HAFI*2
      TUPI=2*PI
      CETRA=PI/180.
      EPS2=EPS/**2
      RIME2=DSQRT(CNE-EPS2)
      IF(SLAT.EQ.0.) SLAT=1.E-5
      FHI1=SLAT*CETRA
      XLA1=DSLCA*CETRA
      IF(DABS(CBLE(5.EQ),SLAT)
      TPSI1=DSIGA(...)
      GO TO
    1 TPSI1=CTAN(FHI1)*RIME2
    2 PSI1=CATAN(TFSI1)
      LCEC=IAES(MCDE).NE.1
      CKA=CAES(LCKA)
      IF(MCDE)2 100,100
C          FCRWARD SCLUTICN.
  100 ALFA1=AZ1*CETRA
      IF(LCEC) ALFA1=TUPI ALFA1=0.
      IF(ALFA1.EG.TUPI) ALFA1=0.
      IF(ALFA1.LT.0.) ALFA1=ALFA1+TUPI
      IF(ALFA1.LE.PI) GO TC 101
      ALFA1=ALFA1-PI
  101 SALFA1=CSIN(ALFA1)
      CPSI1=CCCS(FSI1)*SALFA1
      CALFA1=CATAN(-(CALFA1,TFSI1))
      SIG1=CATAN2(CSIN(PSI1)*SALFA1)
      XLEH1=CARCCS(CFSIW))
      XK=FLA*(CFSIW)
      TWSIG1=TWSIG1+XK*(DSIN(TWSIG1)-XK*(DSIN(2*TWSIG1)/8.))
      CSF=(1.-TWXK)+XK*CKM/BRAVC
      TWSP=TWSIF+DSP
      CSIG=CSF+XK*(XK*5*CSIN(2*CSP)*DCOS(2*TWSP)/8.-DCOS(TWSP)*DSIN(DSF)
     1)
      SIG2=SIG1+CSIG
      SIG=(2*SIG1+CSIG)/2.
      XLB2=CATAN2(DSIN(SIG2),CCCS(SIG2)*CPSIW)
      CLE=(XLE+(SIG2)*SPSIN
      PSI12=CARSIN(CFSIW*PSI2)
      PHI2=CATAN(PSI2)/RIME2)
      XLA2=XLA1+CLE-CPSIW*(CSIG*(1.-R1ME2)-EPS2*XK*(DSIN(2*DSIG)*DCOS(2*
     1PSI2)*XK/(4*CSIN(CSSIG*CCCS(2*SIG))-DSIG*(1.+XK/2.))*CPSIN/4.)
```

165

```
      FLAT=FLTR2/DETRA
      FLCN=XLR4/DETRA
      GC TC ETC
C
C 200  INVERSE SCLUTION.
      XLN2=FLCN*CETRA
      DXLN=XLN-XLN1
      IF(DXLN.GT.PI) DXLA=TLPI-DXLN
      CXLN=DSIGN(CMAX1(DAES(CXLN),CBLE(2.E-4)),DXLN)
      TDLR1=CXLN/2.
      FHI2=FLAT*CETRA
      TPSI2=CTAN(FHI2)*R1ME2
      PSI2=CATAN(TPSI2)
 10   TEMP=CATAN2((TPSI1-TFSI2)/DTAN(HDLB1),TFSI1+TPSI2)
      XLE1=TEMP-F+CLB1
      XLE2=TEMP+F+CLB1
      CLEI=TECLS(XLB1))
      PSIM2=CATAN(TFSI1/CLE1)
      CPSIM2=CCCS(PSIM)
      SIGI1=CATAN2(CFSIM*CSIN(XLB1),CLB1)
      SIGI2=CATAN2(CCPSIM*DSIN(XLB2),CCOS(XLB2))
      SIG=SIGI1+SIGI1/2.
      SIC=SIC-SIGI1/2.
      XK=FLCXK*CCSIN(PSIM)
      TERM=XK*(CCS(2*SIG)*CSIN(DSIG)
     1+DLB2=(CXLA+EPS2*CFSIN*(CSIG*(2./(R1ME2+1.)-XK/2.)-TERM/2.)/2.)/2.
      IF(CABS(TDLR2-HDLB1).LT.1.E-7) GO TO 20
      TDLR1=TDLR2
      GO TC 10
 20   CKM=BRDAVG*(CSIG+TERM-(XK**2)*DSIN(2*DSIG)*DCCS(4*SIG)/8.)/(1.-XK)
      ALFA1=CATAN2(CNE,-CTAN(XLB1)*DSIN(PSI1))
      AZ1=CATAN2(CNE,-CTAN(XLB1)*DSIN(PSI1))
      IF(CKK2.LT.0) AZ1=AZ1+PI
      IF(LLCEG) AZ1=AZ1/CETRA
      IF(DABS(CABS(XLB2)-HAFI).GE.1.E-7) GO TO 35
      ALFA1=ALFA1
      GO TC 36
 35   ALFA2=CAT/A2(ONE,CTAN(XLE2)*SPSI2)
      AZ2=ALFA2
 36   IF(LLCEG) AZ2=AZ2/DETRA
      CKM=CAES(EKM)
      RETURN
      END
C/*
//LINK.USCC CC UNIT=3330,VCL=SER=CISK01,DSN=S2648.ECTLMCD,
//  DISP=SHR,LABEL=(,,,IN)
//LINK.SYSIN CC *
      INCLUDE LSCC(GFCSCT)
```

166

```
ENTRY MAIN
C/*
//GC.FTC1FC01  CC CSN=S2648.GRID2,UNIT=3330,VCL=SER=DISK01,
//             CISP=SHR,LABEL=(,,,IN)
//GC.SYSIN  CE *
-10
    5
  C.  4.    0.
  i  151   15  C.  23
C/*
```

```
C                              GEODST
C*********************************************************************
C*********************************************************************
C
C     SUBROUTINE GEODST IS USED IN PROGRAMS GENBOT AND GRDCHK
C
C     SUBROUTINE GEODST(SLAT,SLON,FLAT,FLON,AZ1,AZ2,DKM,MODE)
C     SOURCE/DATE:  L. R. FRAZO, JAN 1980
C     PURPOSE:  SOLVES FORWARD AND INVERSE GEODESY NAVIGATING
C     TRIANGLE, CALCULATING GREAT CIRCLE DISTANCES, COORDINATES AND
C     AZIMUTHS, TO A GREATER ACCURACY THAN ATTAINED FROM SPHERICAL
C     TRIGONOMETRY.  USES GEODETIC CONSTANTS FROM WGS 72.
C     REFERENCE:  RAND CORPORATION PUBLICATION P-4970.
C     SLAT    LATITUDE OF FIRST STATION
C     SLON    LONGITUDE OF FIRST STATION
C     FLAT    LATITUDE OF SECOND STATION
C     FLON    LONGITUDE OF SECOND STATION
C     AZ1     AZIMUTH FROM STATION 1 TO STATION 2,
C             MEASURED FROM TRUE NORTH
C     AZ2     BEARING, LT 180 DEGREES, MEASURED CLOCKWISE
C             FROM HEADING AT STATION 2 TO LOCAL MERIDIAN
C     DKM     DISTANCE IN KILOMETERS
C     MODE    T: SOLVES INVERSE PROBLEM (REQUIRES
C             SLAT,SLON,FLAT,FLON);
C             F: SOLVES FORWARD PROBLEM (REQUIRES
C             SLAT,SLON,AZ1,DKM)
C             IF IABS(MODE) = 1, AZIMUTHS ARE MEASURED IN RADIANS,
C             OTHERWISE IN DEGREES.
C
C     ALL GEOGRAPHIC COORDINATES ARE MEASURED IN DEGREES
C     AND DECIMALS OF DEGREES.
C     ALL FLOATING POINT NUMBERS ARE DOUBLE PRECISION
C
      IMPLICIT REAL*8(A-H),REAL*8(C-Z)
      LOGICAL LDEG
      DATA EPS/.0818191836/,FRAVO/6356.75023/
      FUNK(SFEI)=CTAN(CATAN(EPS*SPSI/RIME2)/2.)**2
      CNE=1.
      TAPI=CATAN(CNE)*2
```

168

```
      FI=FAFI**2
      TUPI=2*FI
      CETRA=FI/180.
      EPS2=EFS/2*FI
      RIME2=CCSRT(CNE-EFS2)
      IF(SLAT.EC.0.) SLAT=1.E-5
      FHI1=SLAT*EC*CETRA
      XLN1=SLCN*CETRA
      IF(CABS(SLAT).LT.SC.) GO TO 1
      TPSI1=CSIGN(CELE(5.E5),SLAT)
      GO TO 2
    1 TPSI1=(FTAN(FHI1))*RIME2
    2 PSI1=CATAN(TPSI1)
      LLEG=IAES(MODE)
      IF(IAES(MODE).NE.1
      CLK1=CAES(CLKR2)
      IF(MCCE)2CC,10C,1CC
C     FCR AXRC SCLUTION.
  100 ALFA1=AZI
      ALFA1=AZI*CETRA
      IF(CLEG.0.) ALFA1=ALFA1+CETRA
      IF(ALFA1.EC.TUPI) ALFA1=0.
      IF(ALFA1.LT.0.) ALFA1=ALFA1+TUPI
      IF(ALFA1.LE.PI) GO TO 1C1
      CLK4=CLKA-TUPI
      ALFA1=ALFA1-FI
  101 SALFA1=CSIN(ALFA1)
      CPSI1M=CCCS(PSI1)*SALFA1
      CALFA1=CCCS(ALFA1)
      SIG1=CATAN2(-CALFA1,TFSI1)
      XLB1=CATAN2(-CALFA1,CSIN(PSI1)*SALFA1)
      SPSI1M=CSIN(CARCOS(CPSIM))
      XK=FLA1*SIN(SIG1M)
      TWSIG1=TWSIG1+XK*(DSIN(TWSIG1)-XK*(DSIN(2*TWSIG1)/8.))
      CSF=(1.-XKP)*CKM/BRAVC
      TWSF=(1.-XKP)*CKM+CSP
    1 CSIG=CSF+4XK*5*CSIN(2*DSP)*DCOS(2*TWSP)/6.-DCOS(TWSP)*DSIN(DSF))
      1)=CSIG1/2
      SIG2=CSIG1/2.
      SIG12=(2*SIG1+CSIG1)/2.
      XLB2=CATAN2(DSIN(SIG2),CCCS(SIG2)*CPSIM)
      CLEE=XLE1+XLB2
      SFSIG=CCCS(SIG2)*SFSIM
      PSI2=CATAN(CFTAN(PSI2)/RIME2)
      FHI2=CARSIN(CSIN(PSI2)/R1ME2)
      XLA2=XLAN1+CLE-CPSIM*(CSIG*(1.-R1ME2)-EPS2*XK*(CSIN(2*CSIC)*DCOS(4*CSIG
     1PSIG=*XK/(4*DSIN(DSIG)*(CSIN(2*SIG))-DSIC*(1.+XK/2.))*CPSIM/4.)
      FLAT=FHI2/CETRA
      FLCN=XLN2/DETRA
```

```
C
C        GO TO GET A22.
         GO TO 300
C        INVERSE SOLUTION.
200      XLA2=FLLA*DETRA
         CXLA=XLA2-XLA1
         IF(DXLA.GT.PI) DXLA=TLPI-DXLN
         CXLN=DSIG*CMAX1(FAES(CXLN),CBLE(2.E-4)),CXLN)
         FDLB1=CXLA/CETRA
         PHI2=(FL+ATAN(PHI2)*R1ME2
         TPSI2=CTATAN(TPSI2)
         TEMP=CATAN2((TPSI1-TFSI2)/DTAN(HCLB1),TFSI1+TPSI2)
10       XLE1=TEMP+FCLB1
         XLE2=TEMP-FCLB1
         PSIN=CATAN2(CCS(PSIM))
         CPSIM=CATAN2(CCFSIN*CSIN(XLEB1),CLB1)
         VSIG1=CCATAN(TPSI1/CLE1)
         SIG=CCATAN(CPSIM*DSIN(XLB2),DCOS(XLB2))
         VSIG=CSIN*CSIG/2.
         XK=FLNX*XH*CCC*(2*SIG)*CSIN(CSIG)
         TERM=FLNX+SIN(PSIM)
         FCLB1=CFS2*CFSIN*(CSIG*(2./(R1ME2+1.)-XK/2.)-TERM/2.)/2.)/2.
         IF(CAES(FCLB1).LT.1.E-7) GO TO 20
         XLE2=HCLB1-HCLB1)+CSIG))
         FDLB1=+CLE2
20       GO TC 10
         CCKM=ERAVE*(CSIG+TERM-(XK**2)*DSIN(2*DSIG)*DCOS(4*SIG)/8.)/(1.-XK)
         ALFA1=CATAN2(CNE,-DTAN(XLB1)*CSIN(PSTI1))
         AZI=ALFA1+FI
         IF(CKRR.LT.O) AZI=AZI+FI
         AZI1=AZI1/DETRA
30       IF(IDAES(XLB2)-FAFI).GE.1.E-7) GO TC 35
         ALFAC=ALFA1
35       ALFA2=CATAN2(CNE,CTAN(XLE2)*SPSI2)
         AZ2=ALFA2   AZ2=022/DETRA
         IF(LCEG) AZ2=022/DETRA
         CKM=CAES(CKM)
         RETURN
C        LEEUG INIT(FCLB1,CLB1,XK)
         ENC
```

APPENDIX M

```
**********************************************

ECTRACE

**********************************************

**********************************************
C  //ECTRACE JOB (...,2035,,1X82),'ERAZO  SMC1704',TIME=5      ECS00010
C  //FORT.SYSIN DD  REGION.CC=250K                             ECS00020
C  C  * UNIVERSAL MAIN FFCGFAM
C  C  * ESTABLISHES SIZE OF DEPTH MATRIX IN LIEU OF OBJECT
C  C  * TIME LIMITS/CHNL.
C        COMMCN /CIMS/ KMX,KMY                                 ECS00050
C        DIMENSION ZE(151,151)                                 ECS00060
C        KMX=151                                               ECS00070
C        KMY=151
C        CALL TRACER(ZE)                                       ECS00090
C        STOP                                                  ECS00100
C        END                                                   ECS00110

C/*
C  //LINK.USDD DD  UNIT=3330,VOL=SER=DISK01,CSN=S2648.ECTLMOD,
C  //   DISP=SHR,LABEL=(,,,IN)
C  //LINK.SYSIN DD *
C   INCLUDE USDD(TRACER,IDPRCF,NLFRCF,CHNLIM)
C   INCLUDE USDD(CCATAC,BCANG,ANGFRT,IDTSUB,DFEP)
C   INCLUDE USDD(LEGFLT,RETFLT,PACPLT,T2DPLT,ENCFLT)
C   INCLUDE USDD(CRCSCT,SSPPLT,FBLCES)
C   ENTRY MAIN
C/*
C/* REMOVE CR NULLIFY THE FOLLOWING CARD TO PRODUCE PUNCHED SUMMARY
C  //GO.FT07FC01 DD  DUMMY
C  //GO.FT01FC01 DD  CSN=S2648.GFID2,UNIT=3330,VOL=SER=DISK01,
C  //   DISP=SHR,LABEL=(,,,IN)
C  //GO.SYSIN DD *
```

```
CCCCC   REFER TO CHAPTER V 'USER INSTRUCTIONS' FOR A
        DESCRIPTION OF THE FOLLOWING 5 DATA INPUT CARDS.

        TRIAL 84 *** GRABOT GENERATED GRID2 ***
   25.   800.   125.    75.  5000.   150.
   10.     0.   -10.   125.
0155.   125.   -10.   200.   125.     1.
   3    21    10     6

C   FIRST WATER MASS SOUND SPEED PROFILE
NWMR
  500.  1461.   50.  1461.   60.  1447.   70.  1445.   90.  1447.
        1454.3 1050. 1461. 1500. 1470. 4000. 1511.7
C   BOUNDARY BETWEEN FIRST AND SECOND WATER MASS
FENTX  FEXTX   150.
  140.  FEXTY   10.

C   SECOND WATER MASS SOUND SPEED PROFILE
SFWR
  1450.   50.  1462.  200.  1474.5  375.  1474.  750.  1466.
  1465. 1250. 1467. 1590. 4000. 1511.7
C   BOUNDARY BETWEEN SECOND AND THIRD WATER MASS
FENTX  FEXTX   150.
    0.  FEXTY   20.

C   THIRD WATER MASS SOUND SPEED PROFILE
ISCV  2  1470.  5000.  1470.
C/*

LISTING FOR GRDSCT MAY BE FOUND IN G3CP APPENDIX
```

172

```
C*****************************************************************
C                                                               *
C                        TRACER                                 *
C                                                               *
C*****************************************************************
C                                                               *
C     SUBROUTINE TRACER(ZE)                                     *
C     L. R. GERAZC                                              *
C     READS ALL USER INPUT DATA, PERFORMS ACTUAL FAY            *
C   PURPOSE:  TRACING AND OUTPUTS TRACE HISTORY.  INVOKES PLOT ROUTINES. *
C                                                               *
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)
      IMPLICIT LOGICAL*4(L)
      COMMON/CIAS/ KMX,KMY
      COMMON/FRINTS/ FBCT(3C,65C),FANGLE(30,50)
      COMMON/FARAN/ C(5C,3),GRACV(50,3),PIQT,2C(50,3)
      COMMON/FLCTT/ TTL,CAX,CAY,CNLAT,CNLCA
      COMMON/FRCFIL/ OMA3(3),LNOT(2),LEW(2),LNS(2),FENTY(2),FENTX(2),
     1
      COMMON/FACB/ CX,CY,TLT,IRT,IVR,JBT,JTP,JVR,XMIN,YMIN
      COMMON/CTANEL/ CC,CRE,CRN,VV,DEPI,CHS,CHD,
     1ZE,NVALC(3),KUP,KLW,ISSP,KMAX
      DIMENSION TTL(12),ZE(KMX,KMY),OTIT(20)
      DIMENSION XPUN(103),YPUN(103)
      DIMENSION CLC(100),JEL(100),NRB(100),OBB(100),OTT(100),ORM(100)
      DIMENSION FEXTX(2),FEXTY(2),ACT(,FXGAP(2),FYGAF(2)
      DATA CCLEF/'SURF'/,CECTT/'ACT'/,OINIT/'INIT'/,FEND/9999./,
     1CCVER/'CVER'/,CLACR/'UNDR'/,OSTCP/'STCP'/,OCHCK/'
      1
      READ(5CCCC,CTIT
5000  FORMAT(20A4)
      PRINT ECCC,CTIT
6000  FORMAT(//,1CX,20A4,//)
      PLACH ECCC,CTIT
C
C     SEE DCCUMENTATION FOR DESCRIPTION OF INPUT PARAMETERS.
      READ ECICC,XC,ZC,XREC,YREC,ZREC,FREC,ELMAX,FOL
      READ ECICC,XMIN,YMIN,ZCT,ZEND,APST,AREND
      READ ECICC,LCST,HDFNE,FLST,ELENC,HORLIM,SPNGL
      READ ECICC,NLC,NFL,NSTCP,INCPLT,NLEN,ASSF
5010  FORMAT(4F14,3F13)
5011  FORMAT(4CIEIE)
5012  FORMAT(4LE,2I5)
      ELWIC=ELENC-ELST
      IF(ELWIC.LT.0.)  ELWIC=ELWID+360.
```

```
      HDDV=0.
      ELDV=0.
      RHDC=RINQ(AHC,30)
      RNELC=RINQ(AEL,300/NHC)
      NSTGF=PIAC(NSTOP,50)
      IF(NHC.GT.1) HDCV=FLWID/FLOAT(NHD-1)
      IF(NEL.GT.1) ELDV=FLWID/FLOAT(NEL-1)
C
C     READ HEADING OF RAYFANS
      CIRA=FCST+FCDV*FLOAT(NHD-1)/2.
      LCSS=BLFAX.LT.1000
      IF(SMNCL.EC.C.) SMNCL=2.
C
C     FDL DEFAULTS TO TEN WAVELENGTHS.
      WVLN=1.E/FREQ
      IF(FDL.EC.0.) FDL=10000*WVLN
C
C     INPUT WATER MASS DATA
      MAXV=1
      DO 10 J=1,...
      IF(J.LE.NSSP) GO TC 10
      ...V(J)=C
      GO TC...
      LCCAL=.FALSE.
      READ(5)CC5,CMAS(J), KMAX
      AMAXV=MAXQ(MAXV,KMAX)
      READ(5)CIC,SPI,(ZC(I,J),C(I,J),I=1,KMAX)
      READ.EC)C5,...FENTY(J),FEXTX(J),FEXTY(J)
      READ.EC)C5,FENTX(J)-FENTY(J)
      FFYGAF(J)=FEXTX(J)-FENTX(J)
      FXGAF(J)=FYGAP(J).EC.0.
      LENS(J)=FYGAP(J).EC.C.
      LFACT(J)=LEW(J).AND.LNS(J)
      IF(LFACT(J).OR.LNS(J)) GO TO 20
      FAC(J)=FYGAP(J)/FXGAF(J)
20    CONTINUE
C
C     CALCULATE GRADIENTS AND OUTPUT WATER MASS DATA.
      DO 40 J=1,NSSP
      KMAX=AVAL(J)
      KKE=KKMAX-1
      DO 30 I=1,KE
      K=I+1
30    GRADV(I,J)=(C(K,J)-C(I,J))/(ZC(K,J)-ZC(I,J))
40    GRADV(KMAX,J)=.01668
```

```
      PRINT 6040
      DO 50 I=1,MAXV
      IF(I.GT.NVAL(1)) GO TO 41
      PRINT 6041,NVAL(1),CC(I,1), GRADV(I,1)
      IF(I.GT.NVAL(2)) GO TO 42
      PRINT 6042,NVAL(2),CC(I,2), GRADV(I,2)
   41 IF(I.GT.NVAL(3)) GO TO 50
      PRINT 6043,NVAL(3), CC(I,3), GRADV(I,3)
   42 PRINT 6050
   50 CONTINUE
 6041 FORMAT(...,12X,F8.2,5X,FE.2,3X,F8.5)
 6042 FORMAT(...,5X,F8.2,5X,FE.2,3X,F8.5)
 6043 FORMAT(...,50X,F8.2,3X,F8.5)
      PRINT 6060, (CMAS(IB), I=1,NSSP)
      PRINT 6040, (FENTX(I),FENTY(I),FEXTX(I),FEXTY(I),I=1,NFRQ)
 6060 FORMAT('SOUND SPEED PROFILES'//6X,3(13X,'ZC',7X,' C(ZC)'',
     1     7X,'GRAD')//)
 6070 FORMAT(2X,I4,13X)
 6080 FORMAT(3='FRONT/BOUNDARY INTERSECTIONS'//T12,2(27X,'FRONT  ')/T22,
     1  2(8X,'ENTRY',14X,'EXIT')/T23,4(2X,'(',F6.2,',',F6.2,')'))
      LNCR=(XREC.EQ.XO.AND.YREC.EQ.YO)
      2REC=ZREC/100.
      ZO=ZC/100.
C
C     INPUT DEPTH VALUE GRID FROM FILE.
      CALL GRDSC(XMIN,YMIN,XMAX,YMAX,IMAX,JMAX,CHN,ZB)
      FUNCH=CIC.TTL
 7013 FORMAT(4E8./6A8)
      XO=AMAX1(XC,XMIN)
      XO=AMIN1(XC,XMAX)
      YO=AMAX1(YC,YMIN)
      YO=AMIN1(YC,YMAX)
C
C     PULL SOURCE OUT OF WUC.
      CRE=XOO
      CRA=YOO
      WHO=COEF(CRE,CRN,CHA,ZB,IER)
      IF(ZO.GT.WHO)ZC=WHO-100.01
      FICT=CATAN(CBLE(1.1))
      PI = 4 * FICT
      TUFIA = 2 * PI
      CETRA = PI/180.
C
C     MINIMUM DEPTH MUST BE AT LEAST TEN WAVELENGTHS.
      HSTGF=10.*WVLN
C
C     ATTEMPT TO PUT A MISPLACED SOURCE INTO GRID, UNDER WATER.
```

175

```
      DIR=DIR+CETRA                                                     TFA01480
      IF(DIR.GT.FTI) DIR=DIR-TUFI                                       TFA01490
      DIRX=DIR*CUSIN(DIR)                                               TFA01500
      DIRY=DIR*CUCS(DIR)                                                TFA01510
      IF(GTIC.GT.HSTOP.ANC.ZC.GT.O.) GO TC 90                           TFA01520
      CRA=CRA+DIRX                                                      TFA01530
      CRE=CRE+DIRY                                                      TFA01540
   90 WTO=CEEPF(CRF,CRA,ETA,ZB,IER)                                     TFA01550
      IF(IER.NE.C) GO TC 85                                             TFA01560
      ZC=WTC-.001                                                       TFA01570
      GO TC 60                                                          TFA01580
   85 PRINT 6160                                                        TFA01590
 6160 FORMAT(/T25,'***ERRCR*** PRCPAGATION IMPCSSIELE FROM GIVEN SOURCE TFA01600
     1POSITICN AND CEPTH VALUES.'/T25,'RELOCATION ATTEMPTS FAILED.')    TFA01610
      RETUFN                                                            TFA01620
   95 XO=CRE                                                            TFA01630
      YO=CRA                                                            TFA01640
      IF(LACF) ZRFC=ZO                                                  TFA01650
      ZRFC=ZO                                                           TFA01660
C                                                                       TFA01670
      ENTRC FFINT INPUT PARAMETERS.                                     TFA01680
      PRINT 6090                                                        TFA01690
      PRINT 6100, FREQ                                                  TFA01700
      PRINT 6110, XO,YO,ZO                                              TFA01710
      PRINT 6130, NEL,ELST,ELENC,ELDV,NHD,HDST,HDENC,HDDV               TFA01720
      PRINT 6140, XRFC, YRFC, ZRFC                                      TFA01730
      PRINT 6150, KNX,KN,NSTCP  FCL HORLIN, SPNGL                       TFA01740
      PRINT 6150, ELPAX,KN,NSTCP                                        TFA01750
C                                                                       TFA01760
 6090 FORMAT(/T25,'FOLLOWING ADDITICNAL PARAMETERS WERE INPUT:')        TFA01770
 6100 FORMAT(/T25,'SOURCE DATA:'/T25,'FREQUENCY:',F7.2,' HZ.'/)         TFA01780
 6110 FORMAT(/T25,'GRIC CCCRDINATES (',F8.2,',',F8.2,', DEPTH ',        TFA01790
     1F8.2,')')                                                         TFA01800
 6120 FORMAT(T25,'T25,I2,' RAYFANS WILL BE TRACED, FROM ',F5.1,' TC ',F5.1,TFA01810
     1' DEGREES, INCREMENTING EVERY ',F5.1,' DEGREES.'/                 TFA01820
     2T25,I2,' RAYS MAKE A RAYFAN FROM ',F5.1,' TO ',F5.1,' DEGREES TFA01830
     3 INCREMENTING EVERY ',F5.1,' DEGREES.')                           TFA01840
 6130 FORMAT(/T25,'RECEIVER DATA:')                                     TFA01850
 6140 FORMAT(/T25,'GRID DATA:'/T25,'MAXIMUM ARRAY SIZE (',I3,',',I3,')'/TFA01860
     1T25,'DEPTH VALUE= RESCLVEC EY',F4.1,' KM SPACING.')               TFA01870
 6150 FORMAT(T25,'TRACE PARAMETERS:',T25,'PROPAGATICN WILL CONTINUE UNTITFA01880
     1L EACH RAY: ',F5.1,' OR OR ',I4,'. BOTTCM CCNTACTS EXCEECEC.'/    TFA01890
     2T25,'FATH INCREMENT IS ',F6.1,' M.'/T25,'CHANNEL FAILED           TFA01900
     3INGE ',F5.1,' KM. SMALL ANGLE=',F5.2,' DEGREES.')                 TFA01910
      LPLCT=.VALUE.GF.4                                                 TFA01920
      IF(LFLCT)6150  GO TO 100                                          TFA01930
      PRINT 6150                                                        TFA01940
      GO TC 110                                                         TFA01950
```

```
C
C 100  ESTABLISH PLOT PARAMETERS.                                    TFAO1960
      ALEN=PIAC(ALEN,15)                                             TFAO1970
      FACT=ALEN/(15.)                                                TFAO1980
      AXY=AMAX1(15.,AMAX1(XMAX-XMIN,YMAX-YMIN))                      TFAO1990
      ARI5=AXY/(ARI5)                                                TFAO2000
      FDV=AIAC(ARI5)                                                 TFAO2010
      FACT=FACT*AFACT                                                TFAO2020
 7020 PUAC1=TC(C,XMIN,YMIN,FCV,CNX,CNY,CNLAT,CALEN,FACT             TFAO2030
      FORMAT(1CFE.2)                                                 TFAO2040
      ZDV=AIAC(ZENC-ZST)/(3.5)                                       TFAO2050
      IF(ZST.EC.0.) ZST=-.001  ZDV=ZDV+1.                           TFAO2060
      ZEND=ZAINT((ARENC-ARST)/15.)                                  TFAO2070
      ARCV=ZAINT+15.*ZDV                                             TFAO2080
      IF(ARAC(ARENC-ARST,15.).NE.0.) ARDV=ARCV+1.                   TFAO2090
      ARENC=ARST+15.*ARCV                                            TFAO2100
      IF(ARAC(ALEN,FACT).NE.0.) INCPLT)                             TFAO2110
      FRINT 6170,XMIN,ZST,ARST,FDV,FDV,ZDV,ARCV,XMAX,YMAX,ZEND,     TFAO2120
    1 INCPLT=AAC(1,INCPLT)                                           TFAO2130
      FRINT 6200,TI8,'SELECTED.',/T6,'PLOT PARAMETERS:',T37,'X',T47,'Y',TFAO2140
 6170 FORMAT(T57,'Z',/T11,'AXES:'                                   TFAO2150
    2 T25,'START',T67,R./T11E.2,4(F8.2,2X))/T25,'DELTA:',T32,4(F8.2,2X)/T25,TFAO2160
    3 'ENC',T52,4(F8.2,2X)/T13,'SCALING:',T25,'LENGTH',T3,'IN.,  FACTFAO2170
 4TOR=.,FE.4)                                                       TFAO2180
 6180 FCFMAT(/T6.'PLOT CFTICA.')                                     TFAO2190
 6190 FCFMAT(/T6.'TIE,'REFLECTED.')                                 TFAO2200
 6200 FORMAT(/T6.'EACH SEGMENT REPRESENTS',I3,' PCSITION CALCULATICNS.')TFAO2210
C                                                                    TFAO2220
C     INITIAL RECEIVER CISTANCE.                                     TFAO2230
      XDIS=XFREC-X0                                                  TFAO2240
      YDIS=YFREC-YC                                                  TFAO2250
      STR=ATAN2(XCIS,YCIS)                                           TFAO2260
      CFF=ATAN2(XCIS,YCIS)                                           TFAO2270
      IF(DER.LT.-PI) DBP=CBR+TLPI                                    TFAO2280
      LIST=SCRT(XCIS**2+YCIS**2)                                     TFAO2290
      AREC=CIST*CCCS(DBR)                                            TFAO2300
C                                                                    TFAO2310
C     FLCT INITIALIZATICN RCUTINES                                   TFAO2320
      CALL BCAFLT(CFTIT,XMIN,YMIN,FCV,ARST,ARDV,ZST,ZDV,ZEND,       TFAO2330
    1 XFREC,YFREC,AREC,ZFEC,FACT)                                    TFAO2340
    1 CALL TFLT(CIRX,CIFY,CFA,WFO,XO,YO,ZENC,XMIN,XMAX,YMIN,YMAX,ZST,TFAO2350
    1 ZDV,ARST,ARDV,Ze)                                             TFAO2360
      CALL SSFFLT(AVAL,NSSF,CMAS)                                   TFAO2370
      KRAY=0                                                        TFAO2380
      FCL=FCL/1CCC.                                                 TFAO2390
                                                                    TFAO2400
                                                                    TFAO2410
                                                                    TFAO2420
                                                                    TFAO2430
```

177

```
      IEL=0
      SMNGL=SMNGL*DETRA
C
  110 BEGIN NEXT RAYFAN.
      IEL=IEL+1
      IAK=ACC(IEL,13)
      NBNCF=0
      IIC=0
      DO 120  I=1,D
      J=1,J=0.0
      FBC(I,J)=0.0
  120 FANGLE(I,J)=0.0
      ELEVJ=ELST+(IEL-1)*ELCV
      IIC=ELEVC*DETRA
C
CC 130 BEGIN NEXT RAY IN RAYFAN. NBNC COUNTS BOTTOM BOUNCES,
C  NREV COUNTS ALL REVERSALS.
      KRAY=KRAY+1
      ISSP=0
      NBNCC=0
      ACTFL=0
      NREV=-1
      BTWLS=0.0
      LREV=.TRUE.
      LHORZ=.FALSE.
      LLR=.FALSE.
      LBB=.TRUE.
      NPTS=1
      NSEG=-1
      FRANCE=0.
      CP=0.
      EELAPS=0.
      WTHI=WTC
      CEFI=ZC INIT
      CTYPE=CC INIT
      IIC=IIC+1
C
C  INITIAL HEADING.
      AHCG0=HDST+(IHD-1)*HCCV
      IF(AHCG0.LE.0.)   AHCG0=AHDG0+360.
      IF(AHCG0.GT.360.) AHCG0=AHDG0-360.
      CHD(KRAY)=AHCG
      CEL(KRAY)=ELEVO
      THI=WTC
      PHI=AHCG*DETRA
      IF(PHI.GT.FI) PHI=PHI-TUPI
```

```fortran
C     HORIZONTAL DIRECTION VECTOR.                                      TFA02920
      UVX=DSIN(PHI)                                                     TFA02930
      UVY=DCOS(PHI)                                                     TFA02940
C                                                                       TFA02950
C     HORIZONTAL TRACERS.                                               TFA02960
      CRE=XO                                                            TFA02970
      CRA=YO                                                            TFA02980
      CRMIN=10000.                                                      TFA02990
      PRINT 6220,KRAY,IHC,IEL                                           TFA03000
      PRINT 6230                                                        TFA03010
      PRINT 6240                                                        TFA03020
6220  FORMAT('1RAY ',I3,' (',I2,',',I2,') CF RAYFAN ',I3,') TRACE HISTORY'/ TFA03040
     1'0REV ','TIE','X',T24,' DEPTH  GRID         ELEV  BRNG    DIST',   TFA03050
     2T77,'HTR  DEPTH LIMITS  C')                                       TFA03060
6230  FORMAT('+',T108,'GRAZ',2(4X,'BTM'))                              TFA03070
6240  FORMAT(' AC  TYPE  (KM)  (KM)    ')MASS LYR MIN   MAX      (KTFA03100
     1'HEADING ANGLE TO REC  FM REC  TIME ',2(' ',')MASS LYR MIN  MAX   TFA03110
     2M/S)')                                                            TFA03120
6250  FORMAT('+',T197,2(' ANGLE '),'LOSS LPS'/1X,129('-'))             TFA03140
C                                                                       TFA03150
C     HORIZONTAL PLOT POINT.                                            TFA03160
15C   XPLN(NFTS)=CRE                                                    TFA03170
      YPLN(NFTS)=CRN                                                    TFA03180
C                                                                       TFA03190
C     DEPTH DIRECTION VECTOR.                                           TFA03200
      CTI=DCOS(TH1)                                                     TFA03210
      STI=DSIN(TH1)                                                     TFA03220
C                                                                       TFA03230
C     TEST BOTTOM DEPTH AND ITERATE NEXT RAY SEGMENT.                   TFA03240
16C   IF(WH1.LT.STCP.OR.H+1.GE.10.) GO TO 530                          TFA03250
162   CALL ICPRCF(LNUP)                                                 TFA03260
      IF(CLAUF) GO TO 165                                               TFA03270
      IF(LLR) GO TO 170                                                 TFA03280
      IF(.NOT.LBB) GO TO 180                                            TFA03290
C                                                                       TFA03300
C     RAY TAKES ON NEW PARAMETERS FROM NEW WATER MASS                   TFA03310
C     OR A BOTTOM BOUNCE.                                               TFA03320
165   CI=CC                                                             TFA03330
      FDIS=0.                                                           TFA03340
C                                                                       TFA03350
C     VERTEX VELOCITY (RAY INVARIANT).                                  TFA03360
17C   VV=CI/CTI1                                                        TFA03370
      CALL CLRALIA                                                      TFA03380
      G=GRACV(KLP,ISSP)                                                 TFA03390
      CL=FCL
      IF(KLW.LE.KMAX)CL=AMIN1(FDL,(ZC(KLW,ISSP)-ZC(KUP,ISSP)))
```

```
      LRFR=(CT1.NE.0.).AND.(C.NE.0.)
      IF(.NOT.LRFR) GC TC 180
C
C     TRAVEL TIME INTEGRATION LIMITS.
      W2P=C1/C
      W2=W2P
C
C     RADIUS OF CURVATURE.
      RHC=-VV/C
      CTH=CL/RH+C
C
C     MAX THETA INCREMENT IS .001
      CTH=CSIGN(DMIN1(DABS(CTH),DELE(.001)),CTH)
      CCTH=DABS(CTH)
      LLR=.FALSE.
C
C     DISTANCE CALCULATIONS.
180   XDIS=CFE-XREC
      YDIS=CCRN-YREC
      ZDIS=CDFFI-ZREC
      RSC=XDIS**2+YDIS**2
      IF(RSC.NE.C.) GO TC 181
      BTR=0.
      GO TC 182
181   BTR=ATAN((XCIS.YDIS)/CETRA+180.
      IF(BTR.GT.360.) BTR=BTR-360.
      IF(BTR.LE.0.) BTR=BTR+36C.
182   CIST=CSQRT(RSG+(ZDIS**2))
      IF(CINCR.CR.CIST.GE.CRMIN) GC TC 183
C
C     MINIMUM DISTANCE WAS DETECTED.
      CTH-C=A+DG
      CTIME=ELAFS
      CRMIA=CIST
183   IF(CLREV) GC TC 195
      IF(MCC(NSEC,INCPLT).EC.0) GO TO 195
      GO TC 200
C
C     REVERSAL FCINT.
190   NREV=NREV+1
      ELEV=T+1)/CETRA
      PRINT 5040,(IS,SP),NREV,OTYPE,CRE,CRN,DEP1,A+CG,ELEV,BTR,CIST,ELAPS,
     1 GRAT=(IS,SP),KUP,CL,CHC,C1
5040  FCFMAT(1X,I3,1X,A4,2FS.2,F7.3,5F8.3,1X,A4,I4,2(2X,F5.3),F8.5)
      IF(CTYPE.NE.CBC+T) GC TC 185
      ENCL=18C.-EANGLE
      PRINT 5060,FRAZ,RNCL,BTNLSS,NLPS
5060  FORMAT(+,TIC6,FRAZ,2FE.3,F6.1,I3)
```

```
185  IF(RANC.GE.ASTCF.CF.ETNLSS.GE.GE.BLMAX) GO TO 510       TFAC3880
     IF(LBB).FCIS=C.) GO TO 550                              TFAC3890
     IF(LCIS.GE.FCRLIM) GO TO 550                            TFAC3900
     LREV=.FALSE.                                            TFAC3910
     LHCRZ=.FALSE.                                           TFAC3920
     NCR2=NCR2+K2P                                           TFAC3930
190  NCTFL=NCTFL+1                                           TFAC3940
     IF(.NCT.LPLCT) GO TO 200                                TFAC3950
C    PLOT DEFINITION POINT.                                  TFAC3960
     IF(LPLCT) GO TO 191                                     TFAC3970
     XDIS=XCR-X0                                             TFAC3980
     YDIS=YCR-Y0                                             TFAC3990
     RSC=XDIS**2+YDIS**2                                     TFAC4000
     STB=(R/SC.RE.C.) GO TO 192                              TFAC4010
     GO TO 193                                               TFAC4020
191  STB=CTR                                                 TFAC4030
192  STB=ATAN2(XDIS,YDIS)                                    TFAC4040
C    PROJECT SEGMENT ONTO VERTICAL PLOT PLANE.               TFAC4050
193  CIF=STB-CIR                                             TFAC4060
     IF(DIF.LT.-PI) CIF=CIF+TLPI                             TFAC4070
     COSDIF=LCS(DIF)                                         TFAC4080
     RNG=LSCRT(RSC)                                          TFAC4090
     FDEPTH=(ZEANC-CEPI)/ZCV                                 TFAC4100
     FRANGE=(RNG*COSDIF-ARST)/ARDV                           TFAC4110
     CALL RAGFLT(FRANGE,FCFPTH,OTYPE,LBB,IMK)                TFAC4120
C    SORRY, CNLY 200 REVERSALS CR 100 BOTTOM BOUNCES PERMITTED. TFAC4130
200  IF(NREV.EC.200.CR.NFIS.EC.101) GO TO 540               TFAC4140
     LBB=.FALSE.                                             TFAC4150
     IF(LRFF) GO TO 225                                      TFAC4160
C    STRAIGHT RAY SEGMENT.                                   TFAC4170
     CLF=CL                                                  TFAC4180
     TH2=TH1                                                 TFAC4190
     CT2=CTI                                                 TFAC4200
     STC2=STI                                                TFAC4210
     CZ=DLP*CTI                                              TFAC4220
220  CR=DLP*CTI                                              TFAC4230
     GO TO 250                                               TFAC4240
C    REFRACTED RAY SEGMENT.                                  TFAC4250
225  IF(TH1)226,235,226                                      TFAC4260
226  IF(CABS(TH1)-DDTH) 230,235,235                          TFAC4270
C    STCF TH2 AT TURNAROUND POINT.                           TFAC4280
230  TH2=C.                                                  TFAC4290
     CT2=1.                                                  TFAC4350
```

```
            ST2=0.
            IF(ST-CR1) GO TO 245
            GO TO 240
C     NEW ELEVATION ANGLE.
  235       TH2=TH1+DTH2
            ST2=COSH(TH2)
            ST2=DIFFERENCE ... SMALL.
C     DIFFERENCE IN COSINES.  CHOOSE ALTERNATE CALCULATION
C           TH1 ... SMALL.
  240       DCT=CT1-CT2
            IF(DABS(TH1).LE.SMAGL) DCT=(TH2**2-TH1**2)/2.-(TH2**4-TH1**4)/24.
            DZ=R+C*DCT
            DR=R+C*(ST2-ST1)
  245       ...
C     INTERIM DEPTH AND HORIZONTAL COMPONENTS OF RAY SEGMENT HEAD.
  250       CX=DR*LVX
            CY=DR*LVY
            CZ=DR*...
            DEP2=DEP1+CZ
            IF(LBB... GO TO 255
            CEE=CRN+CEX
            CNN=CRN+CEY
  255       IF(LLR) GO TO 265
C     LAYER LIMITS TEST.
            IF(KLP.EG.KMAX) GO TO 260
            IF(DEP2.GT.ZC(KLW,ISSP)) GO TO 450
  260       IF(KLP.EG.1) GO TO 265
            IF(DEP2.LT.ZC(KUP,ISSP)) GO TO 440
  265       IF(LLR) GO TO 300
            IF(LLR2) GO TO 280
C     CHANNEL LIMITS TEST.  CHECK FOR RAY TURNAROUND OR REFLECTION.
C     DEP IS THE BACKLE POINT.
            IF(.NOT.LRFR.OR.(TH2.NE.0.)) GO TO 270
            CEE=CT+V/C  DEP=C+C
  270       IF(DZ.GT.0.) DEP=C+C
            GO TO 420
            IF(DEP2.LT.CHS) GO TO 410
            IF(DEP2.GT.CHD) GO TO 415
C     TEST FOR BOTTOM CONTACT.
  280       WH2=CEEF(CEEF,CNN,DH,ZB,IER)
            IF(IER.NE.0) GO TO 420
            IF(DEP2.NE.WH2) GO TO 350
C     ADVANCE THE TRACE POSITION, TAIL TO HEAD.
  300       HDIS=HDIS+CR
            CRE=CEE
```

```
      CRA=CRA
      CEFI=CEF2
      WH1=WH+WEG+1
      ASEG=NSEG+1
C     TIME CALCULATICN.
  310 IF(CRFR) GC TC 320
      CT=CLF/C1
      GO TO 330
  320 C2=C1+C*CZ
      C1=C2/C
      W12=W2+C2/C
      CT=CLCG(W2*(1.+ST1)/(W1*(1.+ST2)))/G
  330 ELAFS=ELAFS+CT
C     IF AT A REVERSAL FCINT, A DIRECTICN CHANGE IS NECESSARY.
  340 IF(CEB) GC TC 380
      IF(LH+CRZ) GC TC 360
      IF(LREV) GC TO 370
C     NC REVERSAL.
      CT1=CT2
      VT1=VT2
      TH1=TH2
      CTYPE=CCHCK
      GO TC 160
C     TURNAFCLNC.
  360 CT1=1.
      ST1=0.
      TH1=C.
      CTYPE=CCVER
      IF(DZ.GT.0.)CTYPE=CLACR
      GO TC 160
C     SURFACE REFLECTICN.
  370 ST1=-STH2
      TH1=-TH2
      CTYPE=CSLRF
      GO TC 160
C     BCTTCM BCUNCE CHANGES RAY PARAMETERS.
  380 NBAC=NEAC+1
      FBCT(IFL,NENC)=DEP2*1CCC.
      FLEV=TFI/CETRA
      FANGLE(IFL,NENC)=FLEV
```

```
      CALL ECANG(PHI,TH1,CT1,ST1,DHN,UVX,UVY,GRAZ,SLOPE,ZB)        TFA05320C
      AHDG = FH1/DETRA                                             TFA05330C
      IF(AHDG.LE.0.)AHDG=AHDG+360.                                 TFA05340C
      FRAZ=GFAZ/CETRA                                              TFA05360C
      BANGLE=SLCFE/DETRA                                           TFA05370C
      IF(LCSS) BTMLSS=BTMLSS+FBLOSS(FRAZ)                          TFA05390C
      CO=CC1                                                       TFA05400C
      CTYPE=CBCTT                                                  TFA05410C
      NPTS=NFTS+1                                                  TFA05420C
      GO TC 15C                                                    TFA05430C
C                                                                  TFA05440C
C     ECTTCM CCNTACT.   BACKTRACK TO BOTTOM DEPTH.                 TFA05450C
25C   LREV=.TRLE.                                                  TFA05460C
      LBB=.TRUE.                                                   TFA05470C
      LHCR2=.FALSE.                                                TFA05480C
      IF(DFAZ.EC.HT2) GO TC 25C                                    TFA05490C
      CALL CCNTAC(CRE,CRA,CEF1,WH1,CEE,CNN,CEF2,WH2,UVX,UVY,       TFA05500C
     1THI,CRN,ZB,NLFS,EECO)                                        TFA05510C
      CEF=CEF2                                                     TFA05520C
      GO TC 425                                                    TFA05530C
C                                                                  TFA05540C
C     CHANNEL LIMIT REACHED; FORCE A REVERSAL.                     TFA05550C
410   CEP=CHCN/Z20                                                 TFA05560C
415   GCC=CHC                                                      TFA05570C
420   LREV=.TRLE.                                                  TFA05580C
      LEE=.FALSE.                                                  TFA05590C
      LLR=.FALSE.                                                  TFA05600C
      IF=CKZ=CEF.GT.0.                                             TFA05610C
425   IF(DEFI.EC.CEFi GC TC 34C                                    TFA05620C
      CEZ=CEFZ-CEF1                                                TFA05630C
      IF(DEF2=CEF-CEF1                                             TFA05640C
      IF(LLRFR2) GC TC 230                                         TFA05650C
      LLF=CEZ/STI GO TO 43C                                        TFA05660C
      GO TC 220                                                    TFA05680C
C                                                                  TFA05690C
C     BECALSE CF REFRACTICN, RAY HEAD ELEVATION ANGLE CHANGES      TFA05700C
C     AFTER CEPTH CHANGE.                                          TFA05710C
43C   CT2=CTI1-CZ/RHC                                              TFA05720C
      IF(CTZ.GT.1.) GO TC 23C                                      TFA05730C
      TH2=CCSIGA(CARCOS(CT2),CZ)                                   TFA05740C
      ST2=CCSIN(TH2)                                               TFA05750C
      GO TC 245                                                    TFA05760C
C     RAY FENETRATES NEXT SHALLCWER LAYER.                         TFA05780C
440   CEP=2C(KLF,ICSP)                                             TFA05790C
```

184

```
      KUF=KLF-1
      KLH=KLH-1
      GO TO 460
C     RAY PENETRATES NEXT DEEPER LAYER.
  450 DEF=ZC(KLH,ISSP)
      KLF=KLF+1
      KLH=KLH+1
  460 LLR=.TRUE.
      LHCRZ=.FALSE.
      LREV=.FALSE.
      LBB=.FALSE.
      IF(DEP1.EQ.DEP) GO TO 164
      GO TO 426
C
C     TRANSFER TO THE FOLLOWING SECTION ABORTS THE RAYTRACE.
  626 PRINT 6260
 6260 FORMAT(/,1CX,"ERROR AT BOTTOM CONTACT,")
      GO TO ...
  627 PRINT 6270
 6270 FORMAT(/,1CX,"BOTTOM LOSS GATE EXCEEDED,")
      GO TO ...
  628 PRINT 6280
 6280 FORMAT(/,10X,"RAY LEFT GRID,")
      GO TO ...
  629 PRINT 6290,WH2,HSTOP
 6290 FORMAT(/,10X,"BOTTOM DEPTH OF ",F7.3," KM OUT OF ",F7.3," TO 10.0 K
     1M RANGE,")
      GO TO ...
  630 PRINT 6300
 6300 FORMAT(/,1CX,"TRACE DATA EXCEEDS OUTPUT PARAMETERS,")
      GO TO ...
  631 PRINT 6310
 6310 FORMAT(/,1CX,"RAY BECAME TRAPPED IN DUCT OR SOUND CHANNEL,")
  56C CONTINUE
  700 FUNCT 7CCC,ELEVO,ATECO,NFTS,KRAY
 7CCC FORMAT(/,F10.3,2I8)
      FUNCT 7020,(XPUN(I),YFUN(I),I=1,NPTS)
 7020 FORMAT(/,4SEG)
      PRINT 632C,"TERMINATING RAYTRACE AFTER ",I5," SEGMENTS,"
 632C FORMAT(10X,"TERMINATING RAYTRACE AFTER ",I5," SEGMENTS,")
      ACTPL=ACTPL+NEREV
      PRINT 633C,"PLOT BASED ON ",I5," POINTS,",I5,"  REVERSALS AND  "
 633C FORMAT(10X,"PLOT BASED ON ",I5," POINTS,",I5,"  REVERSALS AND  "
     1ITIS," ELTTIC BOUNCES,")
      IF(.ACT.LFLCT) GO TO 570
C
C     PLOT RAY'S HORIZONTAL TRACK.
  57C CALL TGCFLT(XPUN,YPLR,XMIN,YMIN,FDV,NPTS,IMK,KRAY)
      IF(LNGR) GO TO 575
```

```
6340  PRINT 6340,CRMIN,CTIME                                       TFA06280
      FORMAT(10X,'MINIMUM DISTANCE TO RECEIVER',',F8.2,' KM, OCCURS AT TFA06290
     1TIME',F10.3/)                                                 TFA06300
      NREC(KRAY)=NBNC                                               TFA06310
      CBE(KRAY)=CTIME                                               TFA06320
      CDT(KRAY)=CTIME                                               TFA06330
      CRN(KRAY)=CRMIN                                               TFA06340
575   CTYPE=CUSTCRA                                                 TFA06350
      EL=TEL/CETRA                                                  TFA06360
      PRINT 6040,NREV,OTYPE,CRE,CRN,DEP1,AHDG,EL,BTR,DIST,ELAPS,    TFA06370
     1CMAS,CTSF,KUP,CT-C,CT-C,C1                                    TFA06380
      ABNCF=PAXC(NBNCF,NBNC)                                        TFA06390
      IF(DIST.LT.NFC) GO TO 130                                     TFA06400
C                                                                   TFA06410
580   RAY TRACES FOR RAYFAN COMPLETE.                               TFA06420
      IF(NBNCF.EC.0) GO TO 610                                      TFA06430
      NTIMES=NTC/15                                                 TFA06440
      NLEFT=NTC-15*NTIMES                                           TFA06450
6250  PRINT 6250,TEL,ELEVQ                                          TFA06460
      FORMAT('1',5X,'RAYFAN',I3,' COMPLETE.  INITIAL ELEVATION ANGLE = ',TFA06470
     1,F8.2,' DEGREES.'/)                                          TFA06480
      IF(NTIMES.EC.0) GO TO 600                                     TFA06490
      DO 550 J=1,NTIMES                                             TFA06500
      KS=15*(J-1)+14                                                TFA06510
      KE=15*KS                                                      TFA06520
      CALL ANGFFT(KS,KE,NBNCF,CHD,1)                                TFA06530
      CALL ANGFFT(KS,KE,NBNCF,CHD,2)                                TFA06540
550   CONTINUE                                                      TFA06550
600   IF(NLEFT.EG.0)  GO TO 610                                     TFA06560
      KS=15*NTIMES+1                                                TFA06570
      KE=NFC                                                        TFA06580
      CALL ANGFFT(KS,KE,NBNCF,CHD,1)                                TFA06590
      CALL ANGFFT(KS,KE,NBNCF,CHD,2)                                TFA06600
610   IF(TEL.LT.NEL) GO TO 110                                      TFA06610
C                                                                   TFA06620
      ALL TRACES COMPLETE. OUTPUT RAY SUMMARY.                      TFA06630
      PUNCH 7640,FEND                                               TFA06640
      PRINT 6360,KRAY                                               TFA06650
      IF(LNORI)GO TO 640                                            TFA06660
      PRINT 6370                                                    TFA06670
6360  FORMAT('1',I4,' RAY TRACES COMPLETED.'//10X,                 TFA06680
     1'0  MINIMUM DISTANCE TO RECEIVER'/10X,'+ HEADING AT CLOSEST APPRO TFA06690
     2ACH TO RECEIVER'/10X,'* TIME AT CLOSEST APPROACH TO RECEIVER')   TFA06700
6370  FORMAT(//,16X,'RAY',3X,'INITIAL',3X,'INITIAL',1X,'NUMBER OF',5X, TFA06710
     1'  ',5X,'T','NUMBER',2X,'INITIAL',3X,'INITIAL',3X,'NUMBER OF',5X, TFA06720
      FORMAT(13X,'NUMBER',2X,'HEADING',2X,'ELEVATION',1X,'BOUNCES')    TFA06730
6380  DO 620 J=1,KRAY                                               TFA06740
                                                                    TFA06750
```

```
6350    PRINT 6350,J,OHD(J),CEL(J),NRB(J),OBB(J),CTT(J),CRM(J)        TFA06760
        FORMAT(16X,I3,2F10.3,7X,I3,3F10.3)                            TFA06770
6360    CONTINUE                                                      TFA06780
6340    IF(.NOT.LFLCT) RETURN                                         TFA06790
C                                                                     TFA06800
C       PLOT LEGEND, THEN LEAVE PLOT ENVIRONMENT.                     TFA06810
        CALL ENDFLT(SNGL(DIRA),ELST,ELDV,HDWID,NEL)                   TFA06820
        RETURN                                                        TFA06830
        END                                                           TFA06840
C                                                                     TFA06850
CCCCC   SUBROUTINE GRDSCT MAY BE FOUND IN THE G3DF APPENDIX LISTING   TFA06860
                                                                      TFA06870
                                                                      TFA06880
                                                                      TFA06890
```

187

```
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C     ****************************************************
C     *
C     *      SUBROUTINES FOLLOW:
C     *          TCFFCF
C     *          NUPRCF
C     *          CHALIN
C     *          CRESCT
C     *
C     *
C     ****************************************************
C
C
      SUBROUTINE ICFPCF(LNLF)
C
C
C     DETERMINES WATER MASS INDEX BASED ON FRONTAL PENETRATION.
C
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)
      IMPLICIT LOGICAL*4(L)
      COMMON/PROFIL/CMAS(3),LNOT(2),LEW(2),LNS(2),FENTY(2),FENTX(2),
     1 FM(2)
      COMMON/CHANEL/CO,CPE,CRN,VV,DEP1,CHS,CHC,
     1 ZMAX,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
      X=CRE
      Y=CRA
      IPROF=ISSP
      IVSP=ISSP
  100 I=I+1
      IF(LNOT(I)) GO TO 4
      IF(LEW(I)) GO TO 2
      IF(LNS(I)) GO TO 3
      ERND=(Y-FENTY(I))/FM(I)+FENTX(I)
      IF(X.LT.ERND) GO TO 4
  2   IF(Y.LT.FENTY(I)) GO TO 4
      GO TO 1
  3   IF(X.LT.FENTX(I)) GO TO 4
      ISSP=ISSP+1
  1   LNLF=IFFC.NE.ISSP
      KMAX=NVAL(ISSP)
      IF(LNLF) CALL NUPRCF
      RETURN
      END
C
C
      SUBROUTINE NUPRCF
```

188

```fortran
C     CALCULATES NEW WATER MASS PARAMETERS.
      IMPLICIT REAL*8(C-C),REAL*8(G),REAL*8(P-W)
      COMMON/FARAM/C(50,3),GRACV(50,3),PIQT,ZC(50,3)
      COMMON/CHANEL/CO,CRE,CRN,VV,CEP1,CHS,CHC,
     +   ZNRMX,LCAL,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
    1 LOGICAL LCAL
      IF(LCAL(ISSP)) GO TO 15
      DO 10 K=1,KMAX
      C(K,ISSP)=C(K,ISSP)/1000.
   10 ZC(K,ISSP)=ZC(K,ISSP)/1000.
      LCAL(ISSP)=.TRUE.
   15 CONTINUE
C
C     IDENTIFY LAYER
      DO 20 K=1,KMAX
      IF(CEPI.LT.ZC(K,ISSF)) GO TO 30
   20 CONTINUE
      GO TO 31
   30 KM1=K-1
      CO=C(KM1,ISSF)+ (CEF1-ZC(KM1,ISSP))*GRACV(KM1,ISSP)
      KUP=KM1
      KLW=KUP+1
      RETURN
   31 END

      SUBROUTINE CHALIM
C
C     PURPOSE:  CALCULATES REFRACTIVE SOUND CHANNEL LIMITS BASED ON A
C               SNELL'S LAW CONSTANT.  ABSOLUTE LIMITS SURFACE TO 10 KM.
      IMPLICIT REAL*8(C-C),REAL*8(G),REAL*8(P-W)
      COMMON/FARAM/C(50,3),GRACV(50,3),PIQT,ZC(50,3)
      COMMON/CHANEL/CO,CRE,CRN,VV,CEP1,CHS,CHC,
     +   ZNRMX,LCAL,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
    1 GRAC=GRACV(KUP,ISSP)
      GRFR=GRACV(KLW,ISSP)
      IF(GRAC.EC.0.) GO TO 10
      LNEG=GRAC.LF.0.
      LPCS=GRAC.GE.0.
```

```
      IF(LFCS) CHS=DEPI
      IF(LAEC) CHD=DEPI
      IF(.NCT.LFFR) RETURN
      IF(LPCS) GC TO 61
C
C     CALCULATE DEEP LIMIT.
   10 LARGE=VV.GT.1.6
      IF(LARGE) GC TC 55
      IF(KLP.LT.KMAX) GC TC 2C
      IF(GRAC) 55,50,50
   2C IF(C(KLN,ISSP).CE.VV.ANC.LRFR) GC TC 55
      KSTART = KLN+1
      DO 3C K=KSTART,KMAX
      KM1=K-1
      IF(C(K,ISSP).GE.VV.ANC.GRADV(KM1,ISSP).NE.0.) GO TO 40
   3C CONTINUE
      IF(GRADV(KM1,ISSP)) 55,55,35
   35 KM1=KMAX
   4C CHC=ZC(KM1,ISSP)+(VV-C(KM1,ISSP))/GRADV(KM1,ISSP)
      GO TC 100
   5C CHC=CEF1+(VV-CO)/GRAC
      GO TO 100
   55 CHC=10.
      IF(LARGE) GC TO 100
C
C     CALCULATE SHALLOW LIMIT.
   6C LVEF=1 RETURN
   61 IF(C(KLP,ISSP).GE.VV.ANC.LRFR) GO TO 80
      IF(KLP.EC.1) GO TO 1CC
      KENC=KLP-1
      DO 7Q K=1,KENC
      KSTF=K
      IF(C(KSTF,ISSP).GE.VV.ANC.GRADV(KSTP,ISSP).NE.0.) GO TO 90
   7C CONTINUE
      GC CHS=DEF1+(VV-CO)/GRAC
      RETURN
   8C CHS=DEF1+(VV-CO)/GRAC
      RETURN
   9C CHS=ZC(KSTP,ISSP)+(VV-C(KSTP,ISSP))/GRADV(KSTF,ISSP)
      RETURN
  1CC CHS=C.
      RETURN
      END
C
C
      SUBROUTINE CRCSCT(XST,YST,XMAX,YMAX,IMAX,JMAX,DHN,ZE)
```

190

```
      REAL*8 CHA
      COMMON /DIMS/ KMX,KMY
      DIMENSION ZB(KMX,KMY), C10(10)
      X=XST=0..
      X=XMAX=140..
      Y=YMAX=140..
      IMAX=140
      JMAX=140
1000  READ(1,1000) ZB
      FORMAT(10F8.4)
      RETURN
      END
```

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF

```
SUBROUTINES FOLLOW:        ICTSUB
                           CEEP
                           CCNTAC
                           ECANC
                           ANGFRT
                           FELCSS
```

```
C **************************************************

      SUBROUTINE ICTSUB(X,Y,CHN,IER)
C
C     SOURCE: GEORGE GIELIS, ARL
C     PURPOSE: ICS BALLPARK SUBSCRIPTS UNDER HORIZONTAL COORDINATES.
C
      REAL*8 CX,CY,CHN,X,Y,XX,YY
      COMMON /SLBS/CX,DY,ILT,IRT,IVR,JBT,JTP,JVR,XST,YST
      COMMON /EINS/ KMX,KMY,KITR
      XX=X-XST
      YY=Y-YST
      ILT=XX/CTA+1.0
      IRT=ILT+1
      JBT=YY/CTA+1.0
      JTF=JBT+1
      XX=XX-(ILT-1)*DHN
      YY=YY-(JBT-1)*DHN
      IF (XX.LT.YY) GO TC 1C
      IVR=IRT
      JVR=JBT
      GO TO 2C
   10 CONTINUE
      IVR=ILT
      JVR=JTF
   2C IER=C
      IF(ILT.LT.1) IER=IER-1
      IF(JBT.LT.1) IER=IER-3
      IF(IRT.GT.KMX) IER=IER+1
      IF(JTP.GT.KMY) IER=IER+2
      RETURN
      END
```

192

```
C        FUNCTION DEEP(CRE,CRA,CHN,ZE,IER)
C
CCCCC    SOURCE:  GEORGE GIELIS, NRL
C        PURPOSE: GETS WATER COLUMN DEPTH UNDER HORIZONTAL COORDINATES.
         IMPLICIT REAL*8(C-C),REAL*8(G),REAL*8(F-W)
         COMMON /CIMS/ KMX,KMY,KITR
         COMMON /CLBS/CX,CY,ILT,IRT,IVR,JBT,JTP,JVR,XST,YST
         DIMENSION ZE(KMX,KMY)
         CALL CTRLE(CRE,CRA,CHN,IER)
         IF(IER.NE.0) RETURN
         CX=CRE-(ILT-1)*DHN-XST
         CY=CRA-(JBT-1)*DHN-YST
         CC1=ZE(ILT,JBT)
         CC2=ZE(IRT,JBT)
         CV=ZE(IVR,JVR)
         IF(JVR.EQ.JTP) GO TO 10
         CX=(CX-C1)/CHN
         CY=(CY-C2)/CHN
         GO TO 20
   10    CONTINUE
         CX=(C2-CV)/DHN
         CY=(CV-C1)/CHN
   20    CONTINUE
         DEEP=C3*C+CY*CY+D1
         RETURN
         END
C
C        SUBROUTINE CONTAC(CFE,CRA,DEP1,WH1,X2,Y2,Z2,F2,QRE,QRN,
        1 THETA,CELR,DHN,ZE,ALPS,*)
CC       SOURCE:  G. R. ERAZO
CC       PURPOSE: GETS 3D COORDINATES AT BOTTOM CONTACT POINT.
         IMPLICIT REAL*8(A-E),REAL*8(G-H),REAL*8(P-Z)
         COMMON /CIMS/ KMX,KMY,KITR
         COMMON /CLBS/DX,DY,ILT,IRT,IVR,JBT,JTP,JVR
         DIMENSION ZE(KMX,KMY)
CC       INITIAL LOOK STEP HALF OF SEGMENT LENGTH.
         CMOVE=CELR/2.
         TTT=CTA(THETA)
         ALPS=G
         X1=CRE
         Y1=CRA
         Z1=DEP1
```

```
6000  T1=WT1
      IF(IZ1.EC.+1) RETURN
      IF(IZ1.LT.+1) GO TO 5
      PRINT 6000,(X1,Y1,Z1,T+1,X2,Y2,Z2,H2,DELR,DZ,CMOVE,NLPS
      FORMAT(1X,11F10.5,I5)
      X1=X1-CMCVE*CRF
      Y1=Y1-CMCVE*CRN
C     RAY HEAD'S CELL SUBSCRIPTS.
5     I2VR=IVR
      J2VR=JVR
      J2ET=JET
C     COLLECT SUBSCRIPTS AT RAY TAIL.
      T1=DEEF(X1,Y1,DHN,ZE,IFF)
C     RAY TAIL'S CELL.
10    I1VR=IVR
      J1VR=JVR
      J1ET=JET
20    IF(I1VR.AE.I2VR.CR.J1VR.AE.J2VR) GO TC 30
      IF(J1ET.EC.J2ET) GC TC 200
C     RAY TAIL IS NOT IN HEAD'S CELL.
30    NLFS=NALF
      IF(NLPS.GT.50) GO TC 230
C     TAKE CRE STEP FFCM RAY TAIL.  SUBSCRIPTS APPLY TO STEP PCINT.
      XT=X1+CMCVE*CRF
      YT=Y1+CMCVE*CRN
      ZT=Z1+CMCVE*TTT
      T1=DEEF(XT,YT,DHN,ZE,IER)
      IF(INT.CE.+1) GO TC 40
C     STEP PCINT IS ABCVE BCTTCM.  ADVANCE RAY TAIL TC HERE.
      X1=XT
      Y1=YT
      Z1=ZT
      T1=TT
      GO TC 10
C     STEFFEC INTO BOTTCM.  MOVE RAY HEAD TC HERE.
40    X2=XT
      Y2=YT
      Z2=ZT
      H2=HT
C     RAY HEAD'S NEW CELL.  HALVE THE LCCK STEF.
      I2VR=IVR
      J2VR=JVR
      J2ET=JET
      CMCVE=CMCVE/2
      GO TC 20
C     ENTIRE SEGMENT IN SAME CELL.  MOVE RAY HEAD TO CONTACT PCINT.
200   IF(IZ2.EC.+2) RETURN
      CZ=Z2-Z1
```

```
      DR=DSQRT((X2-X1)**2+(Y2-Y1)**2)                              NRFLO148C
      ZC=(T1*Z2+T2*Z1)/((CZ*T1-T2)                                 NRFLO149C
      IF(CZ.NE.0.)GO TO 210                                        NRFLO150C
      CMCVE=(Z1-T1)/(H1-H2)                                        NRFLO151C
210   CMCVE=(Z2C-Z1)/CZ                                            NRFLO152C
222   CMCVE=CMCVE-DR                                               NRFLO153C
      X2=X1+CLACVE*CRN                                             NRFLO154C
      Y2=Y1+CLACVE*CRN                                             NRFLO155C
      H2=CEEP(X2,Y2,DHN,ZB,IER)                                    NRFLO156C
      FETURA                                                       NRFLO157C
230   PRINT*(CCC,X1,Y1,Z1,H1,X2,Y2,Z2,H2,DR,DZ,CMCVE,NLPS          NRFLO158C
      FETURA                                                       NRFLO159C
      ENC                                                          NRFLO160C
C                                                                  NRFLO161C
      SUBRCUTINE BCANG(PHI,TI-ETA,COST,SINT,DHA,UVX,UVY,GRAZ,SLCPE,ZB) NRFLO162C
C     SCURCE:   GEORGE GIELIS, L. R. ERAZC                         NRFLO163C
C     PURFCSE:  CALCULATES NEW RAY DIRECTICN AFTER BOTTOM CONTACT. NRFLO164C
      IMFLICIT REAL*8(C-C),REAL*8(G),REAL*8(F-W)                   NRFLO165C
      COMMCN /CIPS/KMX,KMY,KITR                                    NRFLO166C
      CCMMCN /FARAM/C(50,3),GRACV(50,3),PIQT,ZC(50,3)              NRFLO167C
      CCMMCN/SLES/DX,DY,IL1,IRT,IVR,JBT,JTP,JVR                    NRFLO168C
      CIMEASICA ZE(KMX,KMY)                                        NRFLO169C
C     UNIT VECTOR IA RAY CIRECTION.                               NRFLO170C
      LVR=CCST*LVR                                                 NRFLO171C
      VX=UJVX*LVR                                                  NRFLO172C
      VY=L/VY*LVR                                                  NRFLO173C
      VZ=SINT                                                      NRFLO174C
C     UNIT VECTCR NORMAL TC TRIANGULAR FACET AT BOUNCE PCINT. DX,DY NRFLO175C
C     ARE FARTIAL DERIVATIVES CF Z W/R TC AXES (RATE CF DEFTH CHANGE NRFLO176C
C     W AXES) CALCLLATED IN DEEP. DZ IS MADE NEGATIVE TC PRO-      NRFLO177C
C     JECTC THE NCRMAL UF FRCP THE BOTTOM.                         NRFLO178C
      CZ=-1.0                                                      NRFLO179C
      FNL=CSQRT(CX**2+DY**2+CZ**2)                                 NRFLO180C
      FNX=CX//FNL                                                  NRFLO181C
      FNY=CY//FNL                                                  NRFLO182C
      FNZ=CZ//FNL                                                  NRFLO183C
C     THE SLCPE CF THE FACET IS THE ANGLE THE UNIT NCRMAL MAKES    NRFLO184C
C     WITH THE Z AXIS.                                             NRFLO185C
      SLCFE=CLAFCCS(FNZI)                                          NRFLO186C
      CCT=VX*FNX+VY*PNY+VZ*FNZ                                     NRFLO187C
      GRAZ=CLAFSIN(-CCT)                                           NRFLO188C
      CCA=-2*LCCT                                                  NRFLO189C
      VPX=VX+CCA*FNX                                               NRFLO195C
```

```
      VPY=VPY+CCA*FAY
      VPZ=VPZ+CCA*FAZ
      THETA=CACS*SIN(VPZ)
      VPR=CCCS*(THETA)
      SINT=VFX/VFR
      CCST=VFY/VFR
      UVX=SINT
      LVY=CCST
      PHI=CATAN2(SINT,COST)
      RETURN
      END

      SUBROUTINE ANGPRT(LS,LE,NBOUNC,HEAD,ITYPE)
C
C     SOURCE: GEORGE CIELIS, ARL
C     REVISED: L. R. ERAZO, FEB 1980
C     PURPOSE: OUTPUT OF BOUNCE DATA.
C
      COMMON/FPRINT/FBCT(30,50),KE(50),FANGLE(30,50)
      DIMENSION HEAD(100),KE(50)
      IF(ITYPE.EC.2) GO TO 10
      PRINT 1400
1400  FORMAT(/10X,"ELEVATION ANGLES IN DEGREES BEFORE BOUNCE:")
      GO TO 15
10    PRINT 1500
1500  FORMAT(/10X,"DEPTH IN METERS AT BOUNCE:")
110   PRINT 110,(HEAD(I),I=1,LS,LE)
1100  FORMAT(/1E.,"HEADING:",15F8.2)
      PRINT 1011
1011  FORMAT(3X,"BOUNCE")
      DO 20 I=1,NBOUNC
      KB(I)=I
20    CONTINUE
      IF(ITYPE.EC.2) GO TO 40
      PRINT 120,KE(I),((FANGLE(J,K),J=LS,LE),K=1,I)
120   FORMAT(1X,I6,6X,15F8.2)
30    CONTINUE
40    CONTINUE
      DO 45 I=1,NBOUNC
      PRINT 130,KE(I),((FBCT(J,K),J=LS,LE),K=1,I)
1300  FORMAT(1X,I6,6X,15F8.1)
45    CONTINUE
      RETURN
      END
C
```

```
C
CC
      FUNCTION FBLCSS(ANGLE)                                    FLO02440
C                                                               FLO02450
      DIMENSION ENDS(8),ELCSS(8)                                FLO02460
      DATA ENDS/0.0,11.0,20.0,25.0,35.0,45.0,56.0,90.0/         FLO02470
      DATA ELCSS/0.0,0.0,3.0,4.4,6.7,8.5,10.0,10.0/             FLO02480
      DO 10 I=1,7                                               FLO02490
      IF(ANGLE.GT.ENDS(I).ANC.ANGLE.LE.ENDS(I+1)) GO TO 20      FLO02510
   10 CONTINUE                                                  FLO02520
      IF(I.GT.7) GO TO 3C                                       FLO02530
   20 CONTINUE                                                  FLO02540
      SLCFE=(ELCSS(I+1)-ELCSS(I))/(ENDS(I+1)-ENDS(I))           FLO02550
      FBLCSS=ELCSS(I)+SLCFE*(ANGLE-ENDS(I))                     FLO02560
      GO TO 40                                                  FLO02570
   30 FBLCSS=10.0                                               FLO02580
   40 CONTINUE                                                  FLO02590
      RETURN                                                    FLO02600
      END                                                       FLO02610
                                                                FLO0262C
```

197

```
SUBROUTINES FOLLOW:

          AGAFLT
          BCTFLT
          SWGFLT
          RRAGFLT
          T2DFLT
          EACFLT

      SUBROUTINE BGNPLT(CTIT,XST,YST,FDV,RST,RDV,ZST,ZDV,ZEND,
     1 XREC,YREC,ZREC,FACT)
      REAL*8 TTL(12),CNX,CAY,CAY,CALAT,CNLON
      COMMON /FLC/TTL,CNX,CNX,CKY,CNLAT,CNLCN
      DIMENSION CTIT(20)
      CALL FACT(0.,25.,*FACT,C.,21.*FACT)
      CALL FACT(CR(FACT)
      CALL ...0.,-3)
      CALL ...0.,95.,2)
      CALL ...24.,95.,25.,2)
      CALL ...24.,95.,0.,2)
      CALL ...0.,0.,2)
      CALL SYMECL(1.,20.,6.,12,CTIT,0.,80)
         DRAW HORIZONTAL PLOT ORDER.
      CALL AXIS(0.,0.,'X (KM)',-6,15.,0.,XST,FCV)
      CALL AXIS(0.,0.,'Y (KM)',6,15.,90.,YST,FCV)
      CALL ...0.,15.,2)
      CALL ...15.,0.,2)
      CALL SYMBCL(0.,12,TTL,0.,96)
      XSYP=(XREC+XST)/FCV
      YSYP=(YREC-YST)/FDV
      CALL SYMBCL(XSYP,YSYP,.10,'REC',0.,4)
      XCEA=XCEN-XST)/FDV
      IF(XCEA.LT.0..CR.XCEA.GT.15.) GO TO 10
      YCEA=(YCEN-YST)/FDV
      IF(YCEA.LT.0..CR.YCEA.GT.15.) GO TO 10
      CALL SYMBCL(XCEN,YCEN,.10,12,0.,-1)
```

198

```
      CALL ALPBER(999.,YCEN.,10,CNLAT,0.,5)
      CALL SYMBCL(999.,YCEN.,10,13,0.,-1)
      CALL CNLBER(999.,YCEN.,10,CNLCN,0.,5)
C     CRAB VERTICAL PLCT  B(FDER.
   10 CALL FLCT(CC.,15.5,:3)
      CALL AXIS(0.,15.5,'RANGE (KM)',10,19.,0.,RST,RDV)
      CALL FLCT(-10.,3.5,;2)
      CALL AXIS(C0.,3.5,'DEPTH (KM)',-10,3.5,-9C.,ZST,ZDV)
      CALL FLCT(-10.,0.,2)
      CALL PLCT(1C.,0.,2)
      CALL FLCT(1C.,3.5,2)
      RSYM=(FREC-RST)/RDV
      ZSYM=(ZEAC-ZST-ZREC)/ZDV
      CALL SYMECL(RSYM,ZSYM,.1C,'.REC',0.,4)
      RETLRN
      END
C
      SUBRCUTINE BCTPLT(CTRX,CTRY,DHN,WHO,XO,YO,ZEAD,XMIN,XMAX,YMIN,
     1  YMAX,ZST,ZCV,ARST,ARCV,ZE)
      REAL*E C-N,DTR,WHO,CTRX,TRKX,TRKY
      COMMCN /CTR/ KMX,KMY
      CIMEASICA RTRK(193),TRK(193),IPAT(4),ZE(KMX,KMY)
      CATA IFAT/'C67,E14,1C2E,2C56/
      CTR=CSCFT(CTRX**2+CTRY**2)/ARDV
      HTFK(1)=C..
      RTRK(1)=C..
      TRKX=XQ++RST*CTRX
      TRKY=YC++AFST*DTRY
      TRKX=AMAXI(SNGL(TRKX),XMIN)
      TRKY=AMAXI(SNGL(TRKY),YMIN)
      TRKX=AMIA1(SNGL(TRKX),XMAX)
      TRKY=AMIA1(SNGL(TRKY),YMAX)
      RTRK(2)=C..
      CPTH=CEEP(TRKX,TRKY,C-N,ZB,IER)
      CPTH=AMAXI(CFTH,ZST)
      CPTH=AMIA1(CFTH,ZEAC)
      TRK(2)=(ZEAD-DPTH)/ZCV
   DC 10 I=3,192
      IM1=I-1
      ISTOF=I
      TRKX=TFKX)+CTRX
      IF(TRKX.GT.XMAX.OR.TRKX.LT.XMIN) GO TC 20
      TRKY=TRKY+CTFY
      IF(TRKY.GT.YMAX.CR.TFKY.LT.YMIN) GO TC 20
      FTR=RTRK(IM1)+DTR
```

```
      IF(RTR.GT.15.) GO TO 20
      RTRK(I)=RTR
      CPTH=CEEF(TRKX,TRKY,C+A,ZE,IER)
      CPTH=APAXI(CEPTH,ZST)
      CPTH=APINI(CEPTH,ZENC)
      LTRK(I)=(ZEAD-DPTH)/ZCV
   10 CONTINUE
      RTRK(ISTCF)=0.+RTRK(IMI)
      RTRK(ISTCF)=RTRK(IMI)
      SURL=ZEAC/ZCV
      CALL FLCT(IS.,SURL,2)
      CALL FLCT(C..,SURL,2)
      CALL TCRE(C.,IPAT,-4)
      CALL TCAE(RTRK,LTRK,ISTCF,1)
      RETURN
      END

      SUBROUTINE SSPPLT(NVAL,NSSF,CMAS)

      REAL*8 C,GRADV,PIQT
      COMMON/PARAM/C(50,3),GRADV(50,3),PIQT,ZC(50,3)
      DIMENSICA CP(52),ZCF(52),OMAS(NSSP),NVAL(NSSP)
      XLCC=ZC.
      YLCC=ZC.
      DO 10 I=1,3
      CALL FLCT(I)(XLOC,YLCC,-3)
      IF(I.GT.NSSP) GO TO 10
      CALL SCALE(0..4,.12,CMAS(I),0.,4)
      KMAX=NVAL(I)+KMAX
   10 CONTINUE
      CP(J)=ZC(J,1)
      ZCF(J)=ZC(J,1)
   20 CALL SCALE(ZCB,3.,KMAX,1)
      CALL SCALE(ZCB,6.,KMAX,1)
      KMF1=KMAX+1
      KMF2=KMAX+2
      COVAL=CF(KMF1)
      CDV=CF(KMF2)
      ZVAL=ZCF(KMF1)
      ZDV=ZCF(KMF2)
      CALL AXIS(0..0..'SCUAC SFEEC (M/S)',17.3,0.,CVAL,CDV)
      CALL AXIS(C..,'DEPTH (M)',-9,6.,-90.,ZVAL,ZDV)
   30 ZCF(J)=ZCP(J)
   30 ZCF(J)=1,2,KMAX
      CALL LINE(CF,ZCP,KMAX,1,C,0)
      XLCC=0.
```

```
1C    YLCC=-7.
      CALL FLCT(-20.,10.5,-3)
      RETURN
      END

C
      SUBROUTINE RNGPLT(FRANGE,FDEPTH,OTYPE,LBB,IMK)
C
      LCGICAL LBB
      DATA CINIT/'INIT'/
      IPEN=2
      IF(OTYPE.EC.CINIT) IPEN=2
      IF(FRANGE.LT.0.) IPEN=2
      IF(FRANGE.GT.19.) IPEN=3
      IF(FDEPTH.GT.3.5) IPEN=3
      IF(FDEPTH.LT.0.) IPEN=3
      CALL FLCT(FRANGE,FDEPTH,IPEN)
      IF(OTYPE.EC.CINIT) IPEN=10
      IF(.NCT.LBB.CR.IPEN.EC.3) RETURN
      CALL VYRCL(FRANGE,FDEPTH,.10,IMK,0.,-1)
10    RETURN
      END

C
      SUBROUTINE T2CPLT(XFUN,YFUN,XST,YST,FDV,NPTS,IMK,KRAY)
C
      DIMENSICN XFUN(103),YFUN(103)
      FPN=KRAY
      CALL FLCT(C.,-15.5,-3)
      NPF1=NFTS+1
      NPP2=NFTS+2
      XFLN(NFF1)=XST
      XPLN(NFF1)=XST
      YPLN(NFF1)=YST
      YFLN(NFF1)=FDV
      CALL LINE(XFUN,YFUN,NPTS,1,1,IMK)
      XNCW=XFLN(NFTS)/FDV
      YNCW=YFLN(NFTS)/FDV
      CALL NUMBER(XNOW,YNCW,.1C,FPN,0.,-1)
      CALL FLCT(0.,15.5,-3)
      RETURN
      END

C
      SUBROUTINE ENCPLT(CIRA,ELST,ELDV,HDWID,AFL)
```

```
      CALL FLCT(15,5,-0.6,-3)                              ECP01960
      CALL SYMBCL(0.,0.,.12,'INITIAL ELEVATION ANGLES',0.,24)  ECP01970
      CALL SYMBCL(0.,-.25,.12,'RAYFAN    SYMBOL    THETA',0.,24)  ECP01980
      DO 10 ICL=1,NEL                                     ECP01990
      IMK=MCL(I-1,I+12)                                   ECP02000
      YLCC=-(I-1)*.12+11                                  ECP02010
      FPN=FLC(ITM(I))                                     ECP02020
      CALL ALPNER(.4,YLCC,.12,FPN,0.,-1)                  ECP02030
      CALL SYMBCL(1.5,YLCC,.12,IMK,0.,-1)                 ECP02040
      FPTHET=ELST+(I-1)*ELCV                              ECP02050
      CALL ALPNER(2.3,YLCC,.12,FPTHET,0.,3)               ECP02060
   10 YLCC=YLCC-.5                                        ECP02070
      CALL SYMBCL(0.,YLCC,.12,'HEADINGS',0.,9)            ECP02080
      CALL ALPNER(999.,YLCC,.12,DIRA,0.,3)                ECP02090
      CALL SYMBCL(999.,YLCC,.12,'+/-',0.,5)               ECP02100
      HAMIC=THMIC/2.                                      ECP02110
      CALL ALPNER(999.,YLCC,0.,HAMID,0.,3)                ECP02120
      CALL FLCT(0.,0.,.999)                               ECP02130
      RETURN                                              ECP02140
      END                                                 ECP02150
                                                          ECP02160
```

202

ECCOM

```
************************************************************
*                                                          *
*   PROGRAM ECCOM IS USED TO COMBINE THE PUNCHED HISTORY FROM
*   ECTRACE RUNS TO ALLOW THE INVESTIGATION OF SHADOWING EFFECTS
*                                                          *
************************************************************
```

```
// EXEC FORTCLG
//FORT.SYSIN DD *                                          RECO0010
      DIMENSION XPLN(103),YPLN(103)                        RECO0020
      REAL*8 SIGMA,TTL(12),CNX,CNY,CNLAT,CNLON             RECO0040
      DIMENSION CTIT(20)
      READ 6021,CTIT,TTL,XST,YST,FDV,CNX,CNY,CNLAT,CNLON,FACT
C
C   IF THE USER DESIRES TO CHANGE SIZE OF THIS HORIZONTAL PLOT
C   FROM THAT OF THE FIRST ECTRACE RUN IN THE INPUT DECK, THE
C   SCALING FACTOR, 'FACT' MUST BE CHANGED. THIS VALUE IS THE LAST
C   NUMBER ON DATA CARD #4 OF OF THE FIRST ECTRACE PUNCHED HISTORY.
C   THE VALUE 1 RESULTS IN A 15 INCH PLOT.
C
 6000 PRINT 6000,CTIT,TTL,XST,FDV,CNX,CNY,CNLON,FACT
 6000 FORMAT(10X,2084/10X,6A8/10X,6A8/4X,'XST',6X,'XDV',4X,'YST',5X,'YCV
     1',6X,'CNX',5X,'CNY',3X,'CNLAT',4X,'CNLON'/1X,8F8.3//10X,2(1,INT)
     2AL',/10X,'ELEVATION HEADING   POINTS',5X,'SCALE FACTOR=',F8.2)
      IF(FACT.LE.0.)FACT=1.
      SIZE=FACT*17.
      CALL BINECA(0.,SIZE,0.,SIZE)
      CALL FACTOR(FACT)
      CALL PLOT(0.,-3.,-3)
      CALL PLOT(0.,17.,2)
      CALL PLOT(17.,17.,2)
      CALL PLOT(17.,0.,2)
      CALL PLOT(0.,0.,2)
      CALL SYMBOL(1.,16.6,.12,CTIT,0.,80)
C
C   DRAW POSITION PLOT BORDER.
C
```

RECO0110
RECO0120
RECO0140
RECO0150
RECO0160
RECO0170
RECO0180

```
C

      CALL FLCT(0.,0.,-3) (KM)',-6,15.,0.,XST,FDV)              F2C00190
      CALL AXIS(0.,0.,'X (KM)',-6,15.,0.,XST,FDV)              F2C00220
      CALL FLCT(0.,0.,Y (KM)',6,15.,90.,YST,FDV)              F2C00230
      CALL AXIS(0.,0.,Y (KM)',6,15.,90.,YST,FDV)              F2C00240
      CALL FLCT(15.,0.,2)
      CALL FLCT(15.,0.,2)
      CALL SYMBCL(0.,-6.,.12,TTL,0.,96)
      XCEN=(CAX-XST)/FDV
      YCEN=(CAY-YST)/FDV
      IF(XCEN.LT.0..OR.XCEN.GT.15.) GO TO 102                  F2C00290
      IF(YCEN.LT.0..OR.YCEN.GT.15.) GO TO 102                  F2C00300
      CALL NUMBER(999.,YCEN,.10,12,0.,-1)                      F2C00310
      CALL SYMBCL(XCEN,YCEN,.10,CALAT,0.,5)                    F2C00320
      CALL NUMBER(999.,YCEN,.10,12,0.,-1)                      F2C00330
      CALL NUMBER(999.,YCEN,.10,CALCN,0.,5)
      ELEVG=999.
      GO TO 102

C     PLCT ALL RAYS.

CC
  100 READ(5,5021,END=200)TTT,TTL,XST,YST,FCV,CAX,CNY,CALAT,CALCN,FACT  F2C00360
 5021 FCRMAT(2C44/6A8/10FE.2)
  102 ELEVI=ELEVC
      ELEVC=ELEVO+1.
      READ(50,1)ELEVO,AHDCC,NPTS                               F2C00390
      IF (ELEVC.EQ.9999.) GO TO 100                            F2C00400
 5031 FCRMAT(FIG.3,I5)                                         F2C00420
      PRINT FCRM(5CC3)ELEVO,AHDCC,NFTS                         F2C00430
 5033 FORMAT (EX,2FIG.3,I5)
      READ(50,5041)(XPUN(I),YFLA(I),I=1,NPTS)                  F2C00450
 5041 FCRMAT(10FE.2)                                           F2C00460
      AFF1=AFFTS+1
      APF2=NFFTS+2
      XFUN(NFFF1)=XST
      XPUN(NFFF2)=FCV
      YPUN(NFFF1)=YST
      YFLA(NFFF2)=FCV
      CALL LINE (XPUN,YPLA,NFTS,1,0,0)
      GO TC 102

CCC   END CF FLCT

  200 CALL FLCT(0.,0.,+999)                                    F2C00500
      STOP                                                     F2C00510
      ENC

C     THE FCLLOWING CARCS ARE INSERTED BETWEEN THE PRCGRAM LISTING
```

204

```
C     AND THE DATA.
C     /*
C     //GO.SYSIN DD *
C
C     FOLLOWING THIS CARD THE USER WILL PLACE THE ECTRACE TRIALS TO
C     BE COMBINED IN ONE ECCOM PLOT.
C
C     NO CARD CHANGES ARE NEEDED, HOWEVER IF THE USER DESIRES TO
C     HAVE THE PLOT TITLED ECCOM VICE ECTRACE, CHANGE THE FIRST
C     ECTRACE TITLE CARD TO 'ECCOM TRIAL' FROM 'ECTRACE TRIAL'.
C
C     AT THE END OF ECTRACE DATA HISTORY CARDS PLACE END OF JOB CARD
C*********************************************************************
```

APPENDIX O
FORTCLGW JCL

The job control language (JCL) card for FORTCLGW used in several of
the programs was

    //EXEC FORTCLGW.REGION.GO=250K .

This JCL card identifies and calls the following JCL:

```
XXFORTCLGW PROC DEST=A,IMSL=SF

XXFORT  EXEC  PGM=IEYFORT,REGION=180K

XXSYSPRINT DD SYSOUT=&DEST,DCB=(RECFM=FBA,LRECL=120,BLKSIZE=3360)

SSSYSLIN DD UNIT=SYSDA,SPACE-(CYL,(1,1)),DISP=(,PASS),DSN=&&SYSLIN,

XX   DOE=(RECFM=FB,LRECL=80,BKJSUZE=800)

XXLINK  EXEC  PGM=IEWL,REGION=180K,PARM='MAP,LIST',COND=(4,LT)

XXSYSLIB DD   DISP=SHR,DSN=SYS1.FORTLIB

XX        DD   DISP=SHR,DSN=SYS1.MPSLIB

XX        DD   DISP=SHR,DSN=SYS3.IMSL.&IMSL

XX        DD   DISP=SHR,DSN=SYS1.VTECPLOT

XXSYSLMOD DD DSN=&T(PROGRAM),UNIT=SYSDA,DISP=(,PASS),

XX  SPACE=(CYL,(3,1,1))                         .

XXSYSLIN  DD   DSN=*.FORT.SYSLIN,DISP=(OLD,DELETE)

XX         DD   DISP=SHR,DSN=SYS1.PROCLIE(VMAPP)

XX          DD   DDNAME=SYSIN

XXSYSPRINT DD   SYSOUT=&DEST,DOE=(RECFM=FBA,LRECL=121,BLKSIZE=1210),

XX  SPACE=(CYCL,(1,1)),UNIT=(SYSOUT,SEP=SYSLIN)

XXSYSUT1 DD   UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD,SYSLIB)),

XX  SPACE=(TRK,(10,5))

XXPLOT  EXEC PGM=IEVMAPP,COND=(4LT)

XXSTEPLIB DD   DSN=SYS1.VTECPLOT,DISP=SHR

XXPLOTLOG  DD   SYSOUT=A

XXVECTR1  DD   DISP=(OLD,DELETE),DSN=&&VECTR1

XXSYSVECTR DD   DISP=(OLD,DELETE),DSN=&&VECTR2

XXSYSVECTOR  DD   SYSOUT=(A,,5555)

XXVECTTAPE  DD   DUMMY
```

LIST OF REFERENCES

1. Naval Research Laboratory Report 7815, Horizontal-Gradient Acoustic Ray-Trace Program TRIMAIN, by B. G. Roberts, Jr., 16 December 1974.

2. Harrison, C. H., "Three-Dimensional Ray Paths in Basins, Troughs, and Near Seamounts by Use of Ray Invariants," Journal of the Acoustical Society of America, v. 62, p. 1382-1388, December 1977.

3. Naval Research Laboratory Memorandum Report 3773, Bathymetric Effects on Three-Dimensional Ray Traces, by R. N. DeWitt, April 1978.

4. Naval Research Laboratory Chart, Bathymetry of the Norwegian Greenland and Western Barents Sea, compiled by R. K. Perry and others, 1977.

5. Naval Postgraduate School IBM Scientific Subroutine Package (SSP3), RAIN1 - a set of FORTRAN coded computer subroutines, programmed by R. Hilleary and L. Nolan, July 1974.

6. CONTUR, a FORTRAN-coded computer subroutine, Naval Postgraduate School non-ISML system library, programmed by Bernadette R. Peavey, July 1969.

7. Naval Postgraduate School Technical Note 0141-34, Using the Versatec Plotter at NPS, edited by Sharon D. Raney, December 1979.

8. Clay, C. S., and Medwin, H. Acoustical Oceanography: Principles and Applications, p. 83-87, Wiley, 1977.

9. Naval Oceanographic Office Technical Report 233, Synthetic Bathymetric Profiling System (SYNBAPS), by R. S. Van Wyckhouse, May 1973.

10. Naval Postgraduate School Technical Note 0141-08, Magnetic Tape Usage Under OS/MVP at NPS, by Sharon D. Raney, December 1977.

11. Naval Postgraduate School Technical Note 0141-05, User Libraries and Source Code Editing Under OS, by Sharon D. Raney, July 1977.

12. Naval Postgraduate School non-ISML system library, PTL3D1 - a FORTRAN coded computer subroutine, implemented at NPS by Sharon D. Raney, December 1973.

13. Harrison, C. H., "Acoustic Shadow Zones in the Horizontal Plane," Journal of the Acoustical Society of America, v. 65, p. 56-61, January 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department of Oceanography, Code 68M Naval Postgraduate School Monterey, California 93940	1
4.	Asst. Professor C. Dunlap, Code 68DU Department of Oceanography Naval Postgraduate School Monterey, California 93940	1
5.	Dr. G. Gielis Code 5123 Naval Research Laboratory Washington, D.C. 20390	1
6.	Mr. C. Votaw Code 5123 Naval Research Laboratory Washington, D.C. 20390	4
7.	LCDR R. Christianson Weapons Department (Code 331) Naval Postgraduate School Monterey, California 93940	1
8.	LT L. Erazo ASWOC Sigonella FPO NY 09253	1
9.	Oceanographer of the Navy Hoffman Building No. 2 200 Stovall Street Alexandria, Virginia 22332	1
10.	Commanding Officer Environmental Prediction Research Facility Monterey, California 93940	1
11.	Director Naval Research Laboratory Washington, D.C. 20390	1

12. Scientic Liaison Office, ONR   1
     Scripps Institution of Oceanography
     La Jolla, California   92037

13. Assoc. Professor A. Coppens   2
     Code 61 Cz
     Naval Postgraduate School
     Monterey, California   93940

14. Assoc. Professor R. Shudde   1
     Code 55 Su
     Naval Postgraduate School
     Monterey, California   93940

15. Department of Physics, Code 61   1
     Naval Postgraduate School
     Monterey, California   93940

16. Technical Library   1
     Naval Underwater Systems Center
     Newport, Rhode Island   02840

17. Technical Library   1
     Naval Ocean Systems Center
     San Diego, California   92152

18. Code (08T245)   1
     Naval Material Department
     Navy Department
     Washington, D.C.   20360

19. Code (63R)   1
     Naval Sea Systems Command
     Washington, D.C.   20362